

Open Research Online

The Open University's repository of research publications and other research outputs

Software development as a human activity: a holistic exploration of the social and technical dimensions

Thesis

How to cite:

Tagg, Clare (2001). Software development as a human activity: a holistic exploration of the social and technical dimensions. PhD thesis The Open University.

For guidance on citations see [FAQs](#).

© 2000 The Author



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

Version: Version of Record

Link(s) to article on publisher's website:

<http://dx.doi.org/doi:10.21954/ou.ro.0000e2e3>

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

oro.open.ac.uk



UNRESTRICTED

**SOFTWARE DEVELOPMENT AS A HUMAN ACTIVITY:
A HOLISTIC EXPLORATION OF THE SOCIAL AND
TECHNICAL DIMENSIONS**

Clare Tagg

BA (St Hilda's College, Oxford), MSc (Hatfield Polytechnic, CNAA)

Thesis submitted for the degree of Doctor of Philosophy in Management
September 2000

AUTHOR NO :	M7142456
DATE OF SUBMISSION	: 26 SEPTEMBER 2000
DATE OF AWARD	: 23 MAY 2001

Contents

CDROM	inside front cover
Abstract	
Acknowledgments	
Contents	1
List of figures	4
1 Introduction	7
1.1 Focus of research	8
1.2 Evolution of research question and structure of thesis	11
2 Context	18
2.1 Survey	19
2.2 Literature review	38
2.3 Discussion	52
3 Research paradigm	54
3.1 Approaches in contributing disciplines	55
3.2 Discovering an approach	61
3.3 Research stance	66
3.4 Research quality	73
3.5 Discussion	75
4 Research question	76
4.1 Undertaking longitudinal case studies of software development	76
4.2 Emerging influences in the longitudinal cases	82
4.3 Think tank at Object Technology '95	99
4.4 Discussion	107

5 Research method	109
5.1 Method of analysis	110
5.2 Software support	121
5.3 Development of CDROM	126
5.4 Research quality	143
5.5 Discussion	144
 6 Multinational system description	 145
6.1 Organisational context	146
6.2 Strategy study (Nov 91 - May 92)	151
6.3 Multinational Requirements study (May 92 - Aug 92)	153
6.4 Phase I (Sep 92 - Jun 93)	156
6.5 Use of phase I (Jul 93 - Mar 94)	162
6.6 Phase II (Sep 93 - Mar 94)	167
6.7 Phase II goes live (Apr 94 - Aug 94)	173
6.8 After phase II (Aug 94 - Feb 95)	180
6.9 Discussion	191
 7 Multinational system commentary	 195
7.1 Software conception or inception?	196
7.2 Software construction or evolution?	208
7.3 Software delivery or use?	227
7.4 Discussion	238
 8 Findings	 242
8.1 the Multinational development	244
8.2 Comparison with results from chapter 4	256
8.3 Analysis of the other cases	258
8.4 Discussion	263

9 Conclusions	268
9.1 Evaluation of research approach to longitudinal case studies	270
9.2 Evaluation of key influences on software development	274
9.3 Final thoughts	284
Bibliography	286
Glossary	306
A Questionnaire	309
B Think tank participants	310
C Qualitative analysis software	311
C.1 Analysis stack	312
C.2 Synthesis stack	319
C.3 Pseudonyms stack	322
D Foldouts for chapter 6	323
E Key influences	325

List of figures

Figure 1.1	Structurational model of the influences around software development (extended from Orlikowski & Robey 1991).....	9
Figure 1.2	The symmetrical relation of problem solving activities in software development (extended from Naur 1992 p3)	10
Figure 1.3	Structure of thesis and development of research question	13
Figure 2.1	Distribution of roles of respondents from each course.....	23
Figure 2.2	Purpose and size of respondent organisations	24
Figure 2.3	Purpose of system	26
Figure 2.4	Complexity of system as indicated by hardware.....	26
Figure 2.5	Software used in system	27
Figure 2.6	Size of system in person years	28
Figure 2.7	Involvement of respondents in systems development	28
Figure 2.8	Respondent's rating of factors (% ordered by major + critical)	30
Figure 2.9	Distribution of numbers of factors rated as important.....	31
Figure 2.10	Rating of factors by users, managers and specialists.....	32
Figure 2.11	Significant differences between roles in rating factors.....	33
Figure 2.12	Significant differences between purpose of system in factor ratings	34
Figure 2.13	Significant differences between system complexity (using hardware type as an indicator) in factor ratings.....	34
Figure 2.14	Significant differences between different types of organisation in factor ratings	34
Figure 2.15	Significant differences between different sizes of organisation in factor ratings	34
Figure 3.1	Feedback approach adopted during first year of research.....	63
Figure 4.1	Organisations in the longitudinal cases at the time of the studies.....	78
Figure 4.2	Scope and size of software developments in the longitudinal cases.....	78
Figure 4.3	Development approaches in the longitudinal cases	79
Figure 4.4	Data collected for each study	82

Figure 4.5 Think tank approach	100
Figure 4.6 Categorising factors: the starting point provided in the think tank	101
Figure 4.7 Think tank categories with number of participants who had factors for the category	102
Figure 5.1 Research method used to identify key influences on software development in X	111
Figure 5.2 Major chronological categories and the amount of evidence collected for each category	115
Figure 5.3 Evidence extract 00719 from an interview with Gordan illustrating the trimming of text (text in bold was not trimmed, text in {} were my remarks)	116
Figure 5.4 Major direct influence categories	118
Figure 5.5 Major categories in the commentary on the software development in X.....	119
Figure 5.6 Number of events categorised to each key influence.....	120
Figure 5.7 Relationship between cards in the Hypercard stacks.....	122
Figure 5.8 Navigation symbols used in web pages.....	129
Figure 5.9 Card controlling web page build in Synthesis Hypercard stack	130
Figure 5.10 Structure of the web-site.....	131
Figure 5.11 Story page contains the complete text of the thesis chapter 6.....	132
Figure 5.12 Section 6.6.5 of the thesis as shown on the web-site	133
Figure 5.13 Person page for Matt.....	134
Figure 5.14 Page for evidence 00896, an extract from a conversation with Matt on 14/6/94	135
Figure 5.15 Findings web page	136
Figure 5.16 Page of key influence Belief from section 8.1	137
Figure 5.17 First event categorised to Belief in commentary page for 'Software conception or inception?', section 7.1.4.....	138
Figure 5.18 First synthesis memo for the commentary event 'Small and fast' affecting the choice of process model.....	139
Figure 5.19 Page for social influences showing entry for 'Organisation: Commitment'	140
Figure 5.20 First evidence extract coded to the social influence 'Organisation: Commitment'	140
Figure 5.21 Hyperlinked diagram of the research method allowing access to each aspect of the research in the web-site	142
Figure 6.1 Organisational context, see appendix D.1 for a foldout version of this diagram with a summary of key players.....	147

Figure 6.2 Time line of the Multinational systems development, see appendix D.2 for a foldout version of this diagram	154
Figure 7.1 Key influences on the software development in X	239
Figure 8.1 Establishing the key influences on software development	243
Figure 8.2 Comparison of key influences from analysis of X and results from chapter 4.....	256
Figure 8.3 Impact of key influences from all cases	260
Figure 8.4 Key influences on software development.....	264
Figure C.1 Relationship between the cards in the Hypercard stacks	312
Figure C.2 Scene card from the analysis stack	314
Figure C.3 Recording from scene illustrated in figure C.2	315
Figure C.4 First evidence item associated with recording X60.....	316
Figure C.5 Same evidence card as figure C.4 with trimmed text hidden and categorisation shown.....	318
Figure C.6 Main category card for analysis stack.....	318
Figure C.7 Top level category card for dimension 1 – chronological story	318
Figure C.8 Memo card forming part of the commentary in the synthesis stack.....	320
Figure C.9 Category dimensions in the synthesis stack.....	321
Figure C.10 A sub-category in the commentary classified according to the key influence	321
Figure D.1 Organisational context of Multinational systems development	323
Figure D.2 Time line of the Multinational systems development.....	324
Figure E.1 Effects of key influences in cases	326

1 Introduction

Nearly twenty years ago, Hoare wrote ‘An unreliable programming language generating unreliable programs constitutes a far greater risk to our environment and to our society than unsafe cars, toxic pesticides, or accidents at nuclear power stations’ (Hoare 1981 p199). He could have easily written the same thing today. Many authors (eg Linberg 1999) still begin articles about software by citing some of the well-publicised cases of software failure and problems with computer systems continue to affect adversely everyday life with monotonous regularity.

As an experienced software developer of thirty years with ten years experience of teaching software development, I have been very curious about the unpredictability of project outcomes. Doing everything right in terms of software development method does not guarantee success. Good software is often used in ways that were never intended by the developers. Why? Software professionals, myself included, are perhaps accustomed to the way software development takes on a life of its own outside our control but user management can be dismayed that such a technical discipline is not more deterministic. This research attempts to help developers and users alike have a better understanding of the process of software development.

This chapter defines software development and the focus of interest for the purposes of this research (see section 1.1). During the research, the research question evolved as the research paradigm became established and the research method was developed. Section 1.2 explains how the chapters reflect the evolution of the research question and the contributions of this research to software development and research methods.

1.1 FOCUS OF RESEARCH

It has been argued that software is 'not an ordinary commodity' (Valdez 1988 p4), an 'atypical technology' (Quintas 1994 p29) that has inherent properties of 'complexity, conformity, changeability, and invisibility' (Brooks 1987 p11). Software has many uses (embedded, information systems, education etc), comes in many forms (control system, software package, bespoke system etc) and is developed in many ways. This research is based on the premise that the unique characteristics of software mean that it is worth studying software as a whole irrespective of use, form or origin.

A broad definition of software development is adopted that includes all activities concerned with solving a problem using software. This definition includes deciding what software to write or buy (eg IS/IT strategy, requirements engineering, evaluation), the actual process of building or configuring the software (eg design, programming, testing) and the installation or delivery of the software and associated hardware (eg data conversion, training). It covers the development of software products as well as bespoke systems and includes different types of system from real-time to information-based. Thus this definition includes activities called by some software engineering or systems development (eg Jayaratna 1994).

Orlikowski & Robey (1991) build on the duality of structuration theory that reflects the ways in which social systems both influence and are influenced by human actions, to propose a structurational model of information technology. The model shows the four influences that occur in the interaction between technology and organisations. It has been extended (figure 1.1) to cover the interactions involved when software is not being developed entirely in-house. Although the diagram shows only one user and one developer organisation there may be many; for software products there will be many user organisations and where consultants and external software tools are used there will be more than one developer organisation.

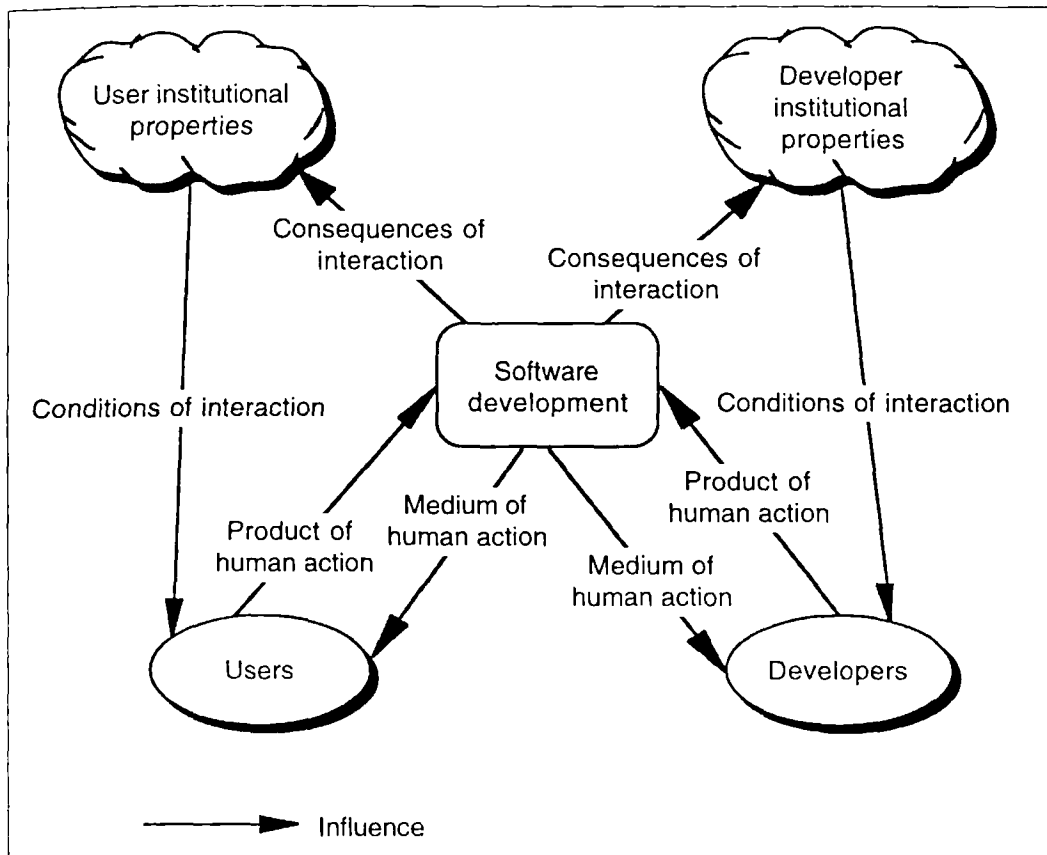


Figure 1.1 *Structural model of the influences around software development (extended from Orlikowski & Robey 1991)*

The model highlights the way in which users and developers have an influence on the outcome of software development (it is a 'product of human action') and this interaction is influenced by their organisation (the 'conditions of interaction'). In addition the software is a 'medium for human action' by both constraining and facilitating user activities, and through this may have an effect on the organisation (the 'consequences of interaction'). As software development is taken to include both product and process, the process is a 'medium for human action' for developers and may impact on the software development process within the developer organisation.

This research focuses primarily on two types of influence: the 'products of human action' and the 'conditions of interaction' while recognising the circular nature of the model (eg using software has an effect on subsequent development of the product). Critically, this means the research is concerned with the influence of actors and organisations on software

development rather more than the influences of software development on organisations or individuals.

Software development covers a variety of activities that have different names depending upon the type of software development environment. Generic names for these activities are used throughout this research to allow all types of software to be discussed. Software development is a problem solving activity and as such is subject to the symmetrical interplay of the three elements: problems, tools and people shown in figure 1.2 (Naur 1992).

This framework is used as a basis for describing the activities of software development.

Conception is the process whereby the purpose and functionality (boundary) of the software are determined; it is analogous to Naur's problem element. **Construction**

includes all aspects of building the software and corresponds to Naur's tools element.

Delivery is about getting the software used, Naur's people element.

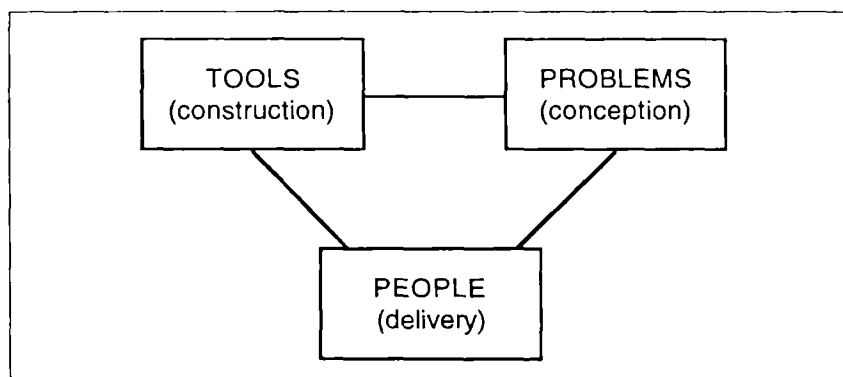


Figure 1.2 The symmetrical relation of problem solving activities in software development (extended from Naur 1992 p3)

The aim of this diagram is to reflect that the three activities can be happening simultaneously and that each needs to be viewed in the context of the other two. It can be contrasted with the more traditional, staged approach illustrated by Sauer's four stage model of information systems development, which was formed from combining innovation and information systems literature (Sauer 1993a p69). Of the four stages initiation corresponds to conception, development to construction, and implementation and operation to delivery.

1.2 EVOLUTION OF RESEARCH QUESTION AND STRUCTURE OF THESIS

Within information systems literature the importance of the human aspects of software development are recognised (eg Avison, Kendall & DeGross 1993) although the emphasis is often on the influences of software development on organisations and individuals. Within the more technical computer science literature the influence of human actors in software development is not ignored but the organisational conditions of interaction are less acknowledged. Weinberg (1971 page vii) believes that programmers would be more *effective if only they were regarded by their managers as a 'human being rather than a machine'*. Naur (1992) has published a collection of his writings dating back to 1965 under the title 'Computing: A Human Activity'. Brooks (1987 p18) in arguing that there is no 'silver bullet' that will solve all the problems of software development suggests, that the 'central question in how to improve the software art centers, as it always has, on people.' Despite this 'most people view software engineering largely as a deterministic *technical* enterprise with some human and social adjuncts, collectively referred to as human factors' (Sharp, Robinson & Woodman 2000 p40).

The emphasis placed on the technical was reflected in the syllabuses of the computer science degree I was teaching in 1993 at the University of Hertfordshire. Not only were the majority of the courses about technical topics but also they were frequently taught with only limited regard to the involvement of people. I taught a second year course on software development covering systems analysis and design. In order to make our course more realistic my colleague and I invented Eagleton Airport complete with people, procedures and a software requirement (Brown & Tagg 1988). We each took on the persona of key characters at the airport so that the students could determine requirements of the proposed system by interviewing people. The course was always launched with an initial interview session involving all 100+ students asking questions in turn, my colleague playing a laid back service engineer complete with glass of beer and me playing one of the management. On one particular occasion, one of the students got particularly frustrated and burst out 'we've been here for an hour but you've still not told us what you want'. This summed up

the problems I felt in trying to educate software engineers. We taught a variety of formal techniques and we tried to put them into a human context but we still gave the impression that software development was essentially a technical business (and many of the students could not wait to get off our course and back on to the real business of programming). This experience led to the initial question for this research:

Does the process of software development depend on more than the traditional concerns of computer scientists?

During the research process the research question has evolved as summarised in figure 1.3 and explained in the following paragraphs.

The question was investigated, as described in chapter 2, using a survey of 769 Open University part-time post-graduate students. Respondents were asked to consider the development of a software system that they had been involved with, and rate how a range of factors contributed to its success or failure. The sample came from students following either a business or a computing course and about equal numbers classified themselves as users, managers or computer specialists. Over a wide range of system types and sizes both technical and non-technical people regarded a number of common factors as critical. Each factor included some human element but only some of them could be regarded as technical. The survey also illustrated some of the problems of using quantitative research in this area.

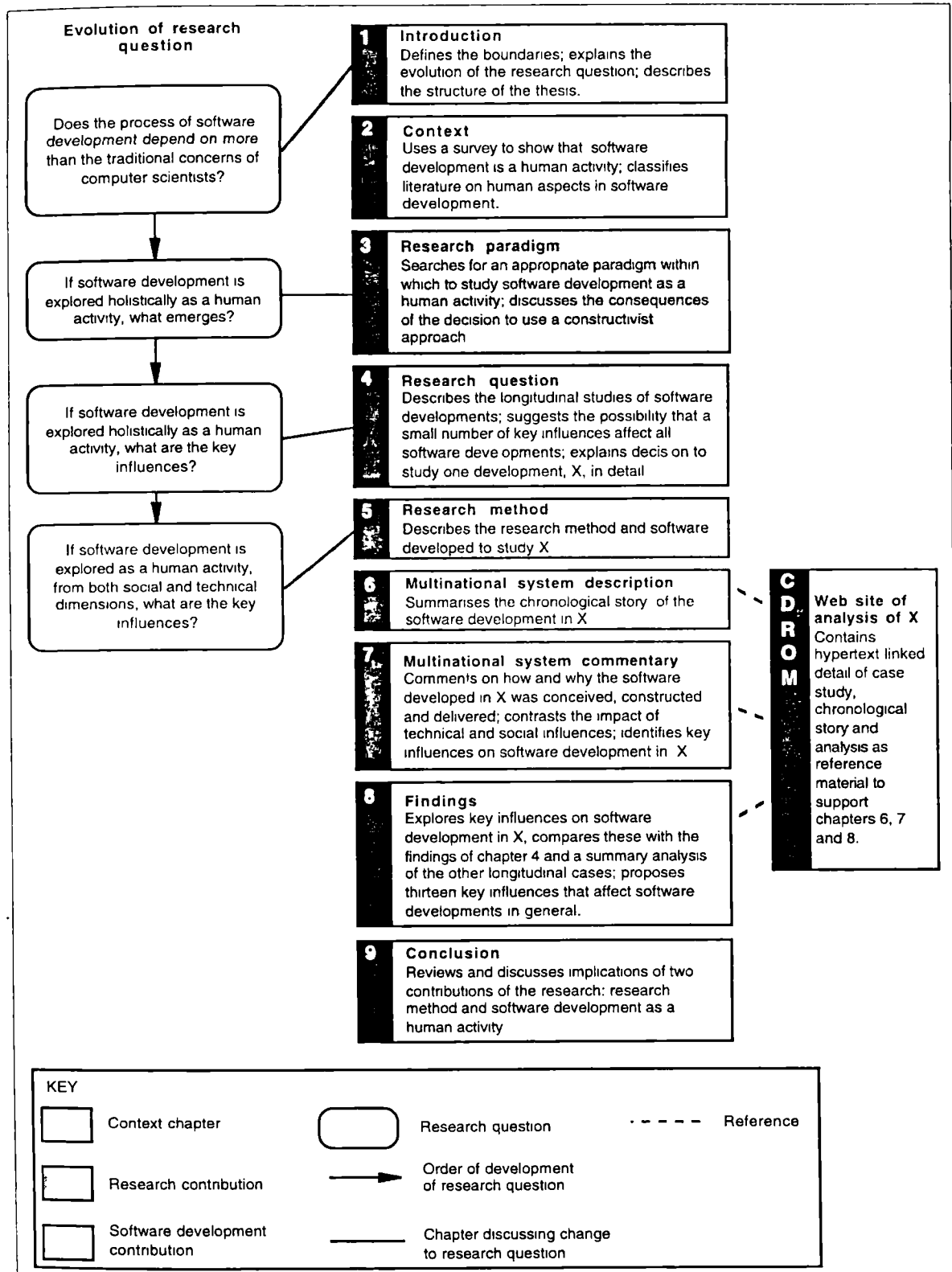


Figure 1.3 Structure of thesis and development of research question

Chapter 2 also demonstrates that the human aspects of software development are treated in a fragmentary way in the literature of computer science, information systems and

management. Five views of software development are identified in the literature; each regards humans as important in software development but for different reasons. Viewing software development as a human activity provides this research with a unifying theme and the work of Checkland on Soft Systems (1981, 1990) shows the usefulness of Human Activity Systems for understanding messy, complex problem situations. As Sharp, Robinson & Woodman (2000 p47) conclude 'if we can really understand what we do, we have some hope of saying what we are and what we should do'.

Viewing software development as a human activity implies a holistic treatment that is problematic for the largely scientific research traditions of the three contributing disciplines. Indeed, Sharp, Robinson & Woodman (2000) suggest using research techniques from sociology and psychology. The struggle to resolve this dichotomy in 1994/5 (described in Tagg 1996a and Open University 1998) led to the adoption of the research stance described in chapter 3. The research stance lies within a constructivist paradigm, which focuses on shared realities that resonate with the interests of both technical and non-technical people involved with software development. The research methodology adopted is based on exploratory longitudinal case studies, which are studied ethnographically and analysed using a grounded approach. This leads to the adoption of the exploratory research question:

If software development is explored holistically as a human activity, what emerges?

Six cases in five organisations were studied longitudinally over about two years as described in chapter 4. The five organisations were chosen because they represented different types of organisation, style and type of software development. The largest project was an organisation-wide administrative system being written in-house for Z, a University. It cost millions of pounds, had about a seven-year development life and was being developed using a rigorous adoption of essentially structured methods for both software design and project management. In contrast the next largest project was being prototyped with some use of structured design methods. The software was being built for X, a subsidiary of a large insurance company, by an information systems department in the parent company. The project was costing £100,000s and involved an international

distributed database. The third project was also a purpose-built information system but was much smaller. It was V's first attempt at building any in-house system that was larger than a standalone PC database. It solved the same kind of problem as Z's project on a much smaller scale, it was being prototyped by relatively inexperienced people and cost £100,000s. Z and V are both educational organisations, but their cultures are quite different. Two projects were followed in the fourth organisation, Y. Both projects were network products with budgets measured in £10,000s, one in the very early stages of development and the other in maintenance. While the developers in Y were amongst the most experienced, their development methods were largely informal. The final and smallest organisation, W, designs and manufactures specialist joinery products. The project followed for W was the introduction of the first computer to the business.

During data collection of the longitudinal studies common themes began to emerge even though the projects were different and at various stages of maturity. This suggested that software developments are influenced by a common, small, set of key influences. A think tank of software professionals confirmed the plausibility of this concept. This led to a refinement of the research question, as discussed in chapter 4, to:

If software development is explored holistically as a human activity, what are the key influences on the conception, construction and delivery of software?

In order to address this question one of the longitudinal studies, X, was selected for detailed analysis. Qualitative methods are not widely used in computer science, information systems and management and methods from other disciplines were deemed inappropriate. Moreover, there is a tradition within qualitative analysis of methods developing alongside the research, so a new method was developed for the analysis of X. The rationale for, and description of this method is contained in chapter 5. The method was supported by new computer software developed by the author and the results are presented as a hyperlinked text on the accompanying CDROM. This CDROM may be read with a web browser on either a Mac or PC. The CDROM allows the reader to explore the richness of the longitudinal case and trace the complete analysis. As explained in chapter 5, this is one of the ways in which research quality is achieved.

The first stage in the research method was to construct a chronological story of the software development told through the words of the stakeholders. The complete chronological story is contained on the CDROM and a summary is presented in chapter 6. The chronological story was analysed for direct influences or absence of influence where it might have been expected. During this process two dimensions emerged often reflecting the way in which the story was interpreted from the perspective of a specialist and of a user or manager. The **social** dimension categorised direct influences that concerned the people and their environments. The **technical** dimension concerned the influences of the actual activities of software development. The categories were grounded in the data but also informed by theory.

The problem of maintaining closeness to data while taking an overall view was addressed by a multi-tier analysis. A synthesis of the effects of the direct influences was written and categorised according to the events affecting *conception, construction and delivery*. From these categories a commentary was written on the events affecting the development process. This commentary is contained in chapter 7 and in hyperlinked form on the CDROM. The events in the commentary in chapter 7 were categorised according to twelve key influences that emerged from the analysis. An assessment of the relative importance of the key influences is used to identify the overall effect of the key influences on *conception, construction and delivery*. Overall this demonstrates that social influences were critically influential at X while technical influences were important but less critical.

Chapter 8 discusses the findings from the analysis of X and describes how an understanding of the importance of the key influences might have affected the development process. To assess the utility of the results, these key influences are compared with the findings from chapter 4 and a summary analysis of the other longitudinal cases. As a consequence a thirteenth influence is added to the key influences on software development.

The thesis concludes (chapter 9) by discussing the contribution to knowledge and critiquing the research. The contribution is in both content and method. The demonstration of the serendipitous nature of software development and the importance of the key influences presented in chapter 8 has important consequences for all those involved with software

development. In adopting a research paradigm unusual for the disciplines involved in this inter-disciplinary project, a research method was devised that properly addressed concerns of research quality and would also be acceptable within those disciplines. This research method included the construction of supporting qualitative data analysis software and techniques for presenting results using hyperlinked web pages.

Finally, a warning: the research method adopted in this thesis is qualitative within a constructivist paradigm and it is quite normal to adopt a personal voice when writing up such research (Meloy 1994). This is not merely a device but reflects the role of the researcher and the reader in the process. I, the researcher, am choosing what to include and what to exclude. With the help of the participants in the longitudinal studies, I have constructed a view of reality. It remains for you, the reader, to be convinced by what I have written and to see reflected within the longitudinal studies similarities with software developments with which you have been involved. There is a problem in that readers may come from traditions that expect the language of research to be passive and objective. For this reason, I have restricted the use of the personal to those areas of this thesis where I felt it was unavoidable. Please, read these sections in the context of the above explanation.

2 Context

This chapter investigates the idea that the process of software development depends on human factors by seeking the opinions of those involved in software development and examining the software development literature.

Software development covers a variety of activities from a simple PC database developed by its user to a complex real-time system running on hybrid technology developed over many years. It consists of the conception, construction and delivery of a software based system and normally involves specialist developers (such as systems analysts, software engineers, programmers), the users of the developed system and various levels of management. Within these stakeholders there is scope for a wide diversity of opinions.

The first part of the chapter (section 2.1) examines the results of a survey that investigated the factors affecting software development. The results show that a wide range of technological and social factors are perceived to influence successful software development. Factors are generally rated similarly by technical and non-technical people or for different kinds of software.

Given that software development has been established for over forty years and the first degrees in computer science began in 1964, one might expect software development to be a cohesive academic discipline. In practice the literature is spread across a number of disciplines. The second part of the chapter (section 2.2) examines human aspects of software development in literature from the disciplines of computer science, information systems and management.

The chapter concludes (section 2.3) that software development is a human activity, that human concerns are not treated holistically in the literature and that surveys are not a good way of further investigating the research question.

2.1 SURVEY

A survey was used to test the hypothesis that although technology is important in software development, there is a range of human factors that are also important. The survey also investigated whether the factors regarded as critical to success vary according to the perspective of the respondent, type of system or organisation.

A questionnaire was used because it is a relatively cheap means to get quickly to a diverse audience. The target was technical and non-technical people who had recently been involved in the development of a software-based system. Open University part-time post-graduate students from the MBA (Masters of Business Administration) and CCI (Computers in Commerce and Industry) programmes provided an appropriately varied audience. The responses came from a wide variety of organisations, technical/non-technical respondents, and related to different types of systems (see section 2.1.3) with no one group dominating. Problems of interpretation with such a diverse audience were considered in the design and piloting of the questionnaire.

2.1.1 Development of the questionnaire

In developing the questionnaire (see Appendix A), a number of key decisions were taken based on published checklists for questionnaire design (eg Tull & Hawkins 1980). The overall layout of the questionnaire was kept clean and uncluttered, data entry codes were avoided and it was restricted to four sides of A4 to encourage people to respond.

The questionnaire begins with an explanation for the context of the research in order to encourage non-technical people involved in software development to complete it. The term software development was avoided because of its technical connotations and replaced by 'development of systems that include computers'. Unfortunately, there was an emphasis in the questionnaire on software systems for organisations reflecting the state of the research question at the time. This emphasis did not seem to cause problems with completing the

questionnaire for respondents developing products incorporating software, but may have reduced the number of responses in this area.

The first sections of the questionnaire contain questions about the respondent and their organisation. In order to establish the frame of reference within which the questionnaire should be answered, the respondent was asked to think about their role in software development. This frame of reference was further refined by asking respondents to describe the most recent system they had been involved in developing or enhancing. The aim was to encourage respondents to think about a specific system when answering the main part of the questionnaire and so reduce the chance of getting 'received wisdom' responses.

The main part of the questionnaire covered factors affecting the success of the development of that system. It was designed as multiple choice to get a measure of the importance of a wide range of factors. The list of factors was derived from the researcher's experience enhanced by the responses in the pilots. The factors were deliberately chosen to be about different things, to be at different levels and listed in an adhoc order to encourage respondents to think widely about the reasons for success or failure. The factors were presented on the questionnaire in the order shown below:

- Technical competence of the computer specialists
- Involvement of users of the system in the design
- Commitment of computer specialists to the system
- Commitment of users to the system
- Commitment of managers to the system
- Computer resources available during development
- Methods used by computer specialists
- Widespread consultation
- Amount of democracy in organisation
- Amount of realistic planning used in development
- Consideration of users' jobs
- Way in which development team is organised
- Financial health of the organisation
- Monitoring and control of development
- Experience of the developers
- Size of development team
- Experience in the organisation of this kind of development

- Computer equipment (hardware)
- Computer programs (software)
- Design of the computer screens and reports (user interface)
- Testing
- Introduction of the system
- Establishing requirements
- Organisational politics
- Factors external to the organisation
- Training
- Organisational change (now or later)

A four point ordinal scale (none, minor, major, critical effect) was used to polarise responses to these factors and discourage indiscriminate choice of the middle ground. The aim was to establish the most important factors, not all factors that affected the development. A no opinion option and comment space were included to allow respondents the option of ignoring or re-interpreting factors they were not clear about. Such rating scales can be very unreliable because terms are not interpreted consistently, respondents try to please or the list of factors is incomplete (Oppenheim 1966). The wording in the questionnaire was carefully considered during its design and piloting. Respondents appeared to treat the questionnaire seriously: very few did not describe a particular system, most responses showed considerable variation over the rating of the factors (see figure 2.9) and many (over 350) explanatory comments were added. Respondents were given the option of adding their own factors and a relatively small number did this (see section 2.1.6) but this survey was not trying to be exhaustive, just to demonstrate the range of factors that are important.

The questionnaire was piloted to assess the ease of use, evaluate the interpretation by different types of respondents and to get as comprehensive a list of factors as possible. The first pilot was conducted in person with four people (including technical and non-technical) to ensure the overall shape of the questionnaire was appropriate. This piloting affected the use of multiple choice and open-ended questions as well as providing refinement for the instructions.

The second pilot was sent to twelve people known to the researcher who are involved in software development within different industries. Half the people were users rather than developers and their level of technical knowledge varied. The questionnaire was well

received although only half were returned. Follow up of non-responses revealed that the respondents had been too busy to complete the questionnaire. The responses represented people with different roles in software development and varying levels of technical knowledge. The main changes arising from the pilot were an increase in the number of factors and the inclusion of a uniform way for respondents to indicate effects that were positive or negative.

2.1.2 Distribution of the questionnaire

During the summer of 1993, the questionnaire was distributed at residential school to 220 Open University Business School MBA students studying one of the core courses. The response rate was good (55%) but only 67% of these completed the whole questionnaire. From informal conversations at the schools, non-completion of the whole questionnaire seemed to arise mainly because of lack of time. On reflection, residential schools are probably not a good opportunity for distributing questionnaires as the students are very busy and they already have to complete a course evaluation questionnaire.

The problem of non-completion of the second part of the questionnaire was puzzling. It was not clear whether respondents failed to answer the questions because they had no involvement in a software development, because their involvement had only been as users, or they were too busy to complete the questionnaire. A follow-up was mailed to the ten people who had given their name and address. The result of this was inconclusive (only three were returned - one just completed the questionnaire but did not explain why and two had no involvement in development even using a broader definition than implied by the original questionnaire).

In consequence, the questionnaire was slightly modified to make it clear that users involved in development should complete the last part of the questionnaire. It was mailed in September 1994 to 549 students who were taking or had recently taken the project management or project course as part of the Computers in Commerce and Industry Scheme. The response rate was 35% and the numbers completing the second part of the questionnaire much better (84%) probably as a consequence of the higher number of

software development specialists. A comparison of the returns from the two courses is shown in figure 2.1.

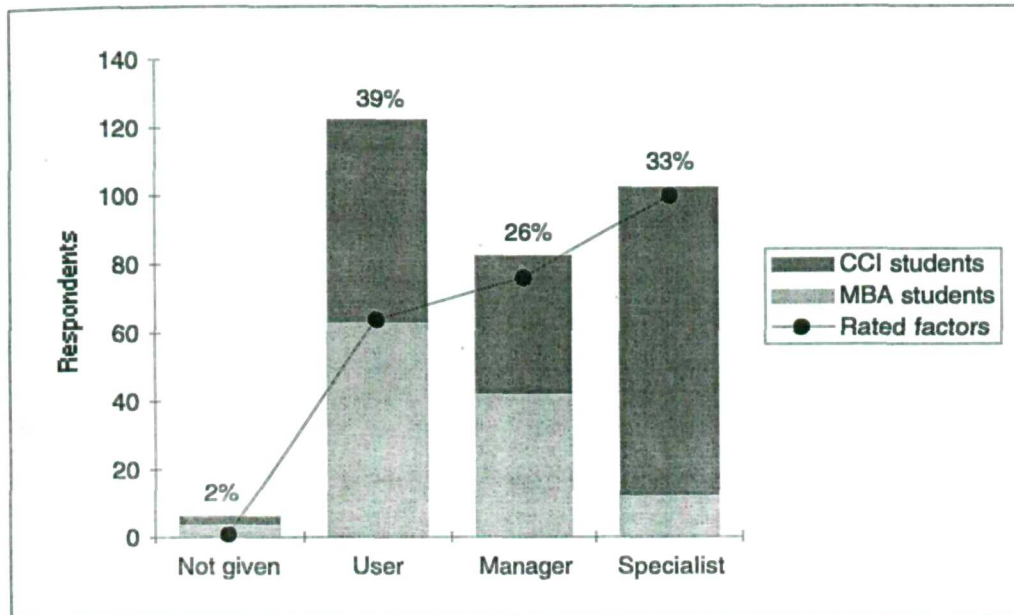


Figure 2.1 Distribution of roles of respondents from each course

In this figure, the respondent's role in development was categorised based on their responses to the first question:

- user, for those who only use the system;
- manager, for those who are responsible for the whole system or the computer system, and may use the system (but are not computer specialists);
- specialist, for those who are computer specialists (they may additionally be users or managers)

Those who specified an 'other' role have been individually categorised using the above definitions as a basis. The line 'rated factors' shows the proportion over both courses who completed the full questionnaire.

Responses from those people who completed the full questionnaire on both courses (241 or 31% of total sample) were combined and analysed using SPSS¹ (Cramer 1994) for the quantitative data and NUD*IST² for the qualitative data. The first set of analyses in section 2.1.3 shows the characteristics of the population. The analysis of the ratings for the factors for the whole population is contained in section 2.1.4. Section 2.1.5 investigates whether the responses varied for different types of respondent, organisation or system.

2.1.3 Characteristics of the population

The responses came from a range of different organisations although there was a preponderance of responses from manufacturing and service industries as shown in figure 2.2. This also shows that small organisations were not well represented except within the service sector.

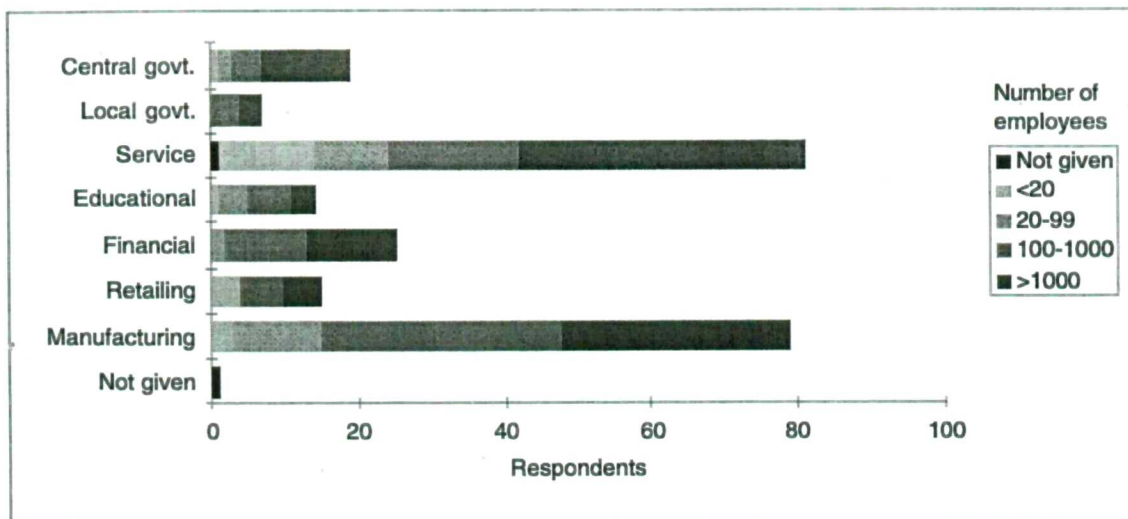


Figure 2.2 Purpose and size of respondent organisations

As might be expected from the size of the organisations, there was a high level of computerisation. In 68% of organisations most people used a computer in their work

¹SPSS - Statistical Package for the Social Sciences developed by SPSS Inc (www.spss.com)

²NUD*IST – Non-numerical Unstructured Data: Indexing, Searching and Theorising, a qualitative data analysis package developed by QSR International (www.qsr.com.au)

(although in about half of these the use varied), 58% had both mainframe or mini computers as well as PCs and electronic mail was used by many people in 54% of organisations.

The systems described on the questionnaire (and for which the respondents rated the factors) varied considerably in both purpose and size. There were very small projects involving things like the installation of standard software on a PC network, or the development of a system to match teachers with teaching posts. Examples of slightly larger projects were the installation of a PC-based email system or the provision of a barcode interface to a mainframe. Larger still were projects such as the installation of EPOS tills for public houses and the provision of management information. There were also very large projects such as the development of a personnel and payroll system, new microprocessor or a computer game simulation environment.

Based on experience from the pilot, respondents were asked to describe the system in free-form responses to questions about its purpose, hardware and software, length of development and number of people involved. This information was coded into categories covering the type of the system, hardware, software and size in person years. A broad-brush approach was used because of the variability of detail and technical knowledge of the respondents. As shown in figure 2.3, four types of systems were identified:

Edutainment	education, training, publishing or games but excluding education management
Real-time/control	process control, communications, embedded applications, CAD/CAM
Information	operational data-based systems such as stock control, financial management and management information systems
Systems/R&D	installation of network systems, email applications and various investigatory projects

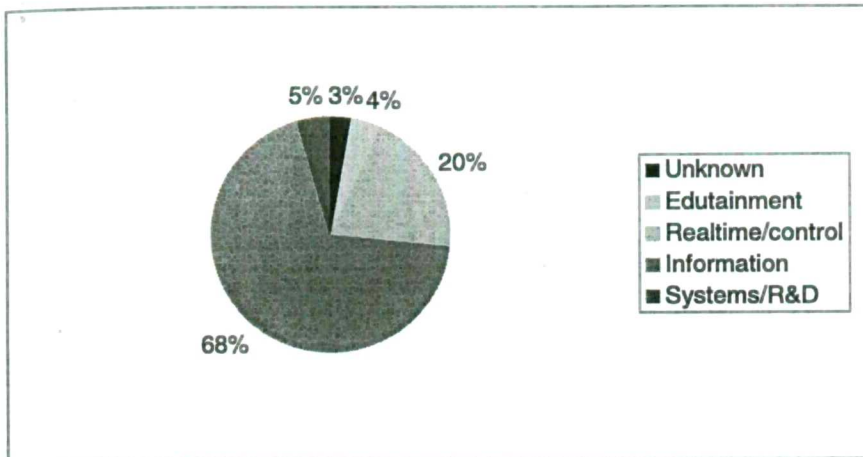


Figure 2.3 Purpose of system

Hardware was categorised on a four point scale of increasing complexity: standalone PC systems (single concurrent user), PC networks (many users but little interaction), mainframe/mini (many interacting users) and hybrid (typically client-server or communication systems). The systems were fairly evenly spread amongst these four categories as illustrated in figure 2.4.

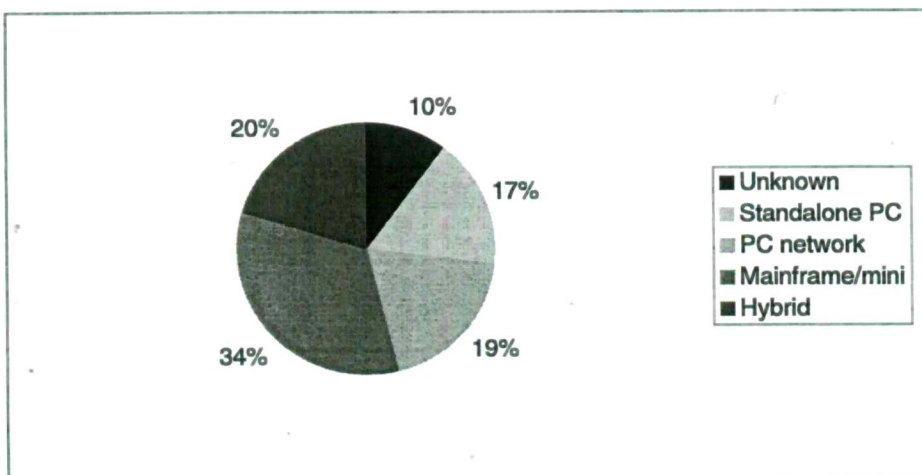


Figure 2.4 Complexity of system as indicated by hardware

In many cases, it was not possible to categorise the software used for the system or the size of the system. In the case of software, this was largely because the respondent did not know. In addition, for large systems several types of software were often listed. Figure 2.5 shows the range of software used in the systems; the large specialised category includes problem-specific packages (eg payroll, CAD) as well as CASE tools. As a consequence of the

preponderance of information systems it is perhaps not surprising that database or 4GL products were mentioned nearly twice as often as programming languages.

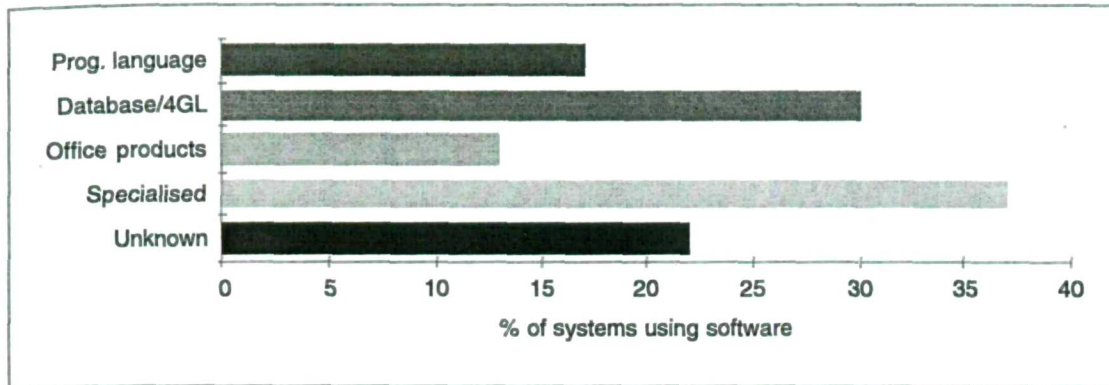


Figure 2.5 Software used in system

The final aspect of the systems, their size is summarised in figure 2.6. This figure was prepared by multiplying the length of the development in years by the average number of full-time (or equivalent) people working on the development. The large number of unknowns arises either because the respondent was not aware (or in one case 'too embarrassing to publish') how many people were involved or the project was ongoing. There was also some difficulty over whom to include as only some respondents carefully spelt out the amount of time given by different classes of people (developers, system support, users etc.). The length of the projects ranged from 0.5 person months (a PC installation) to 7500 person years (telephone exchange processing). In general the very large projects involved large numbers of people (500 people worked on developing the telephone exchange system) and took several years to develop. The longest running development was a command and control system supporting defence aircraft which was started in 1980 and was not due to be finished until at least 1997. Figure 2.6 shows the range of project sizes in the sample.

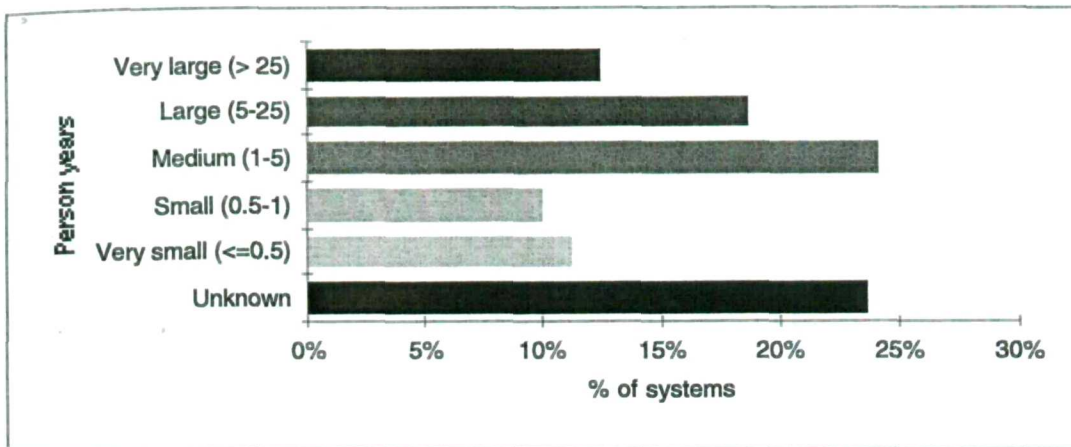


Figure 2.6 Size of system in person years

The final chart in this section, figure 2.7 shows the variety of ways in which the respondents were involved in the development of the system. It is interesting to note that managers were often involved in all aspects of development.

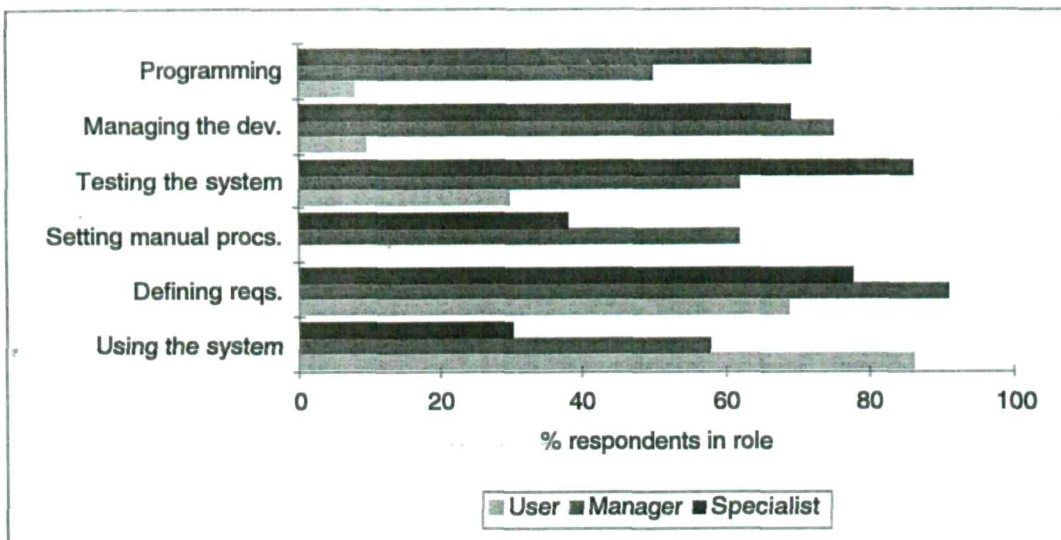


Figure 2.7 Involvement of respondents in systems development

In general there was considerable interest shown by respondents in the research. Additional comments were added by many (37%), respondents frequently explaining why factors were important or unimportant in their particular system (see section 2.1.6). Interest in the results of the research resulted in 132 respondents providing a mail address. A number of respondents provided telephone numbers and offered further assistance.

2.1.4 Factors important for software development

The main aim of the survey was to demonstrate that for a wide variety of systems there are many factors, both technological and human, which are important for successful software development. Figure 2.8 shows the % of respondents who rated each factor. The graph has been ordered to show those factors rated major or critical by most people. Throughout subsequent analysis, these two ratings are combined and described as important to allow for different interpretations of the words 'critical' and 'major' by respondents. Please note, the description of the factors has been abbreviated for presentation (see section 2.1.1 for full text).

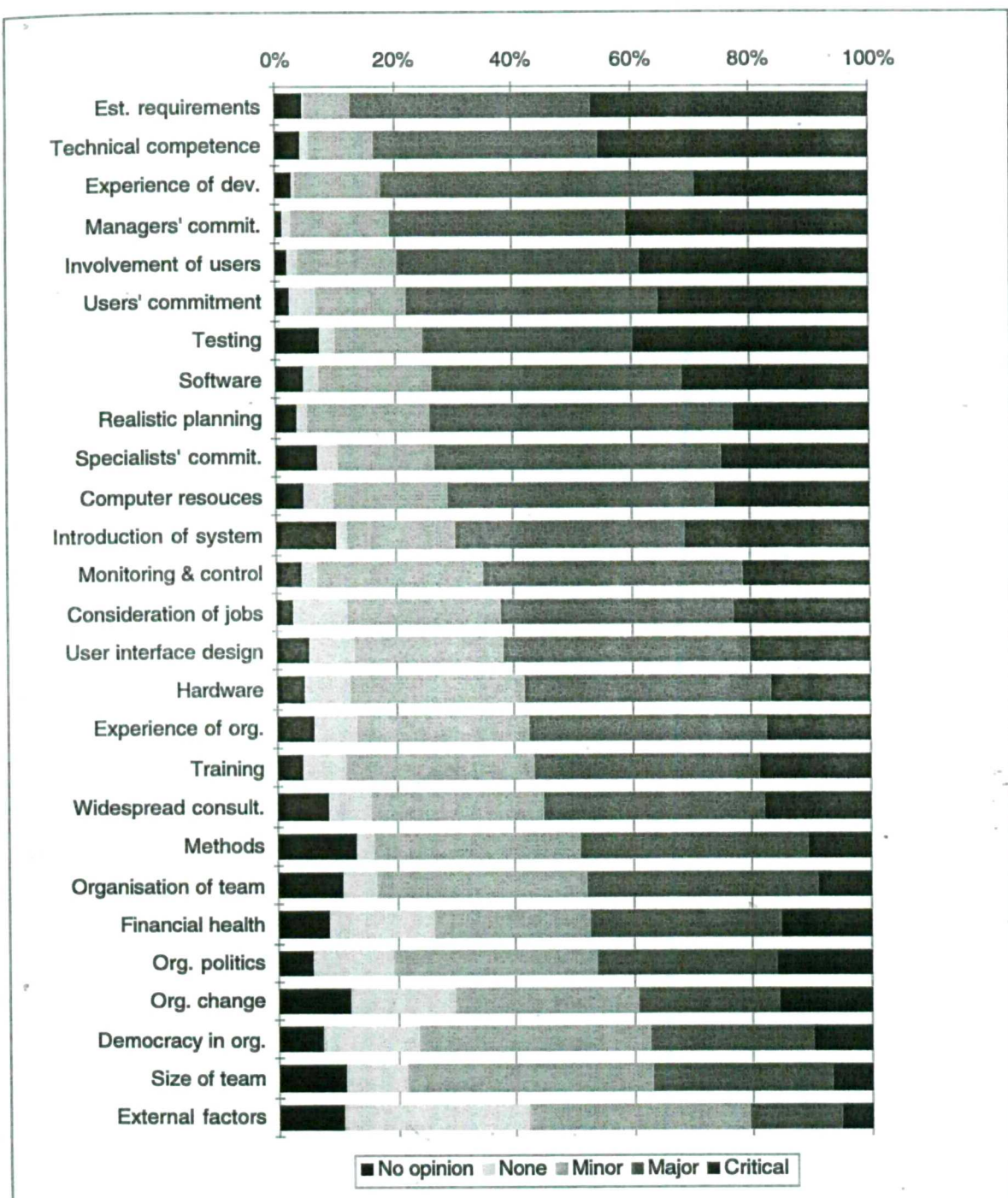


Figure 2.8 Respondent's rating of factors (% ordered by major + critical)

The striking feature of this graph is that 19 of the factors were regarded as important by over half the respondents and even the factor at the bottom of the chart (external factors) was important for over 20% of respondents. There are no real surprises in the factors topping the list but for software development specialists it is perhaps surprising that issues such as user interface design and methods come so low down. This raises the question is

there a difference in how technical and non-technical people rate the factors; this is addressed in section 2.1.5. Section 2.1.6 discusses the comments made by respondents about these factors and additional factors they identified as important.

The next figure (2.9) shows the number of factors regarded as important by each respondent. As might be expected, the distribution is normal (at 5% significance for kurtosis and skewness) with a mean of 16.7 and standard deviation of 4.8.

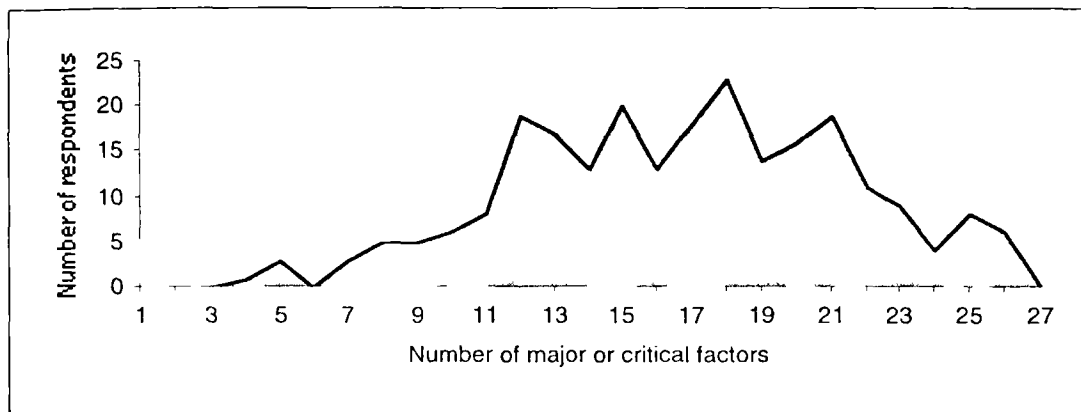


Figure 2.9 Distribution of numbers of factors rated as important

The rating of factors as particularly positive or negative was less satisfactory and meant that this aspect of the responses could not be analysed. Only 37% of respondents indicated positive or negative for any factor and there was some inconsistency between the use of the sign as indicated by accompanying comments. It was not clear whether a '+' indicated:

 this factor was well-done and so contributed to the success of the project

 or

 if only this factor had been well-done, it would have contributed to the success of the project.

People seemed to use the signs to reflect strong feelings; although organisational politics was regarded as important by relatively few people (46%), it attracted the most '+' or '-' signs (63 respondents). This confusion over the '+' indicates a weakness in the design of the questionnaire which existed despite changes following piloting.

2.1.5 Stratification

This section examines significant differences in the rating of factors between respondents based on their role, organisation or system. In these comparisons, respondents who recorded a 'no opinion' have been excluded. The Kruskal-Wallis 1-way anova test corrected for ties (Siegel & Castellan 1988) was used to test for significant differences.

Figure 2.10 compares the rating by the three roles (user, manager and specialist) against the overall rating. There is a remarkable degree of agreement with specialists rating slightly more factors as important on average (17.3 specialist, 16.3 user, 16.1 manager).

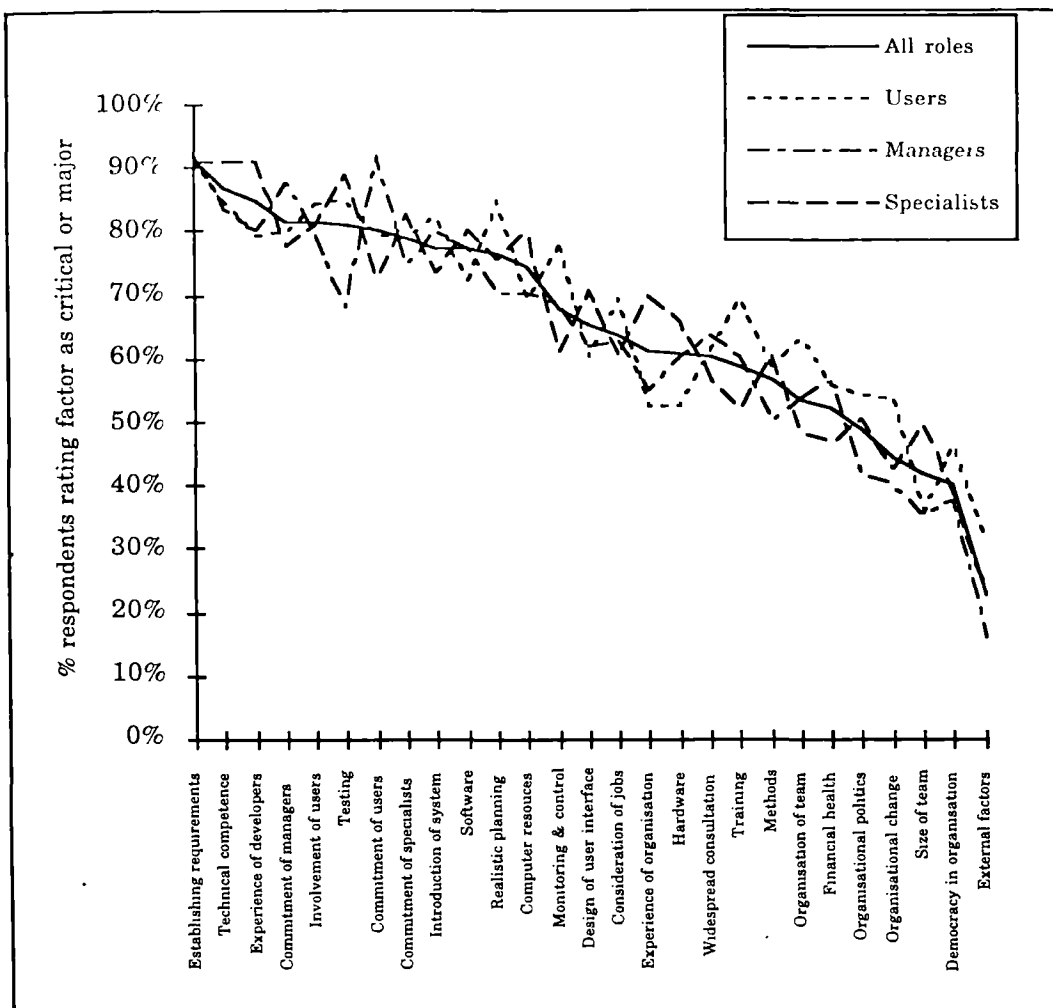


Figure 2.10 Rating of factors by users, managers and specialists

The factors where there is a difference (at 5% significance) on Kruskal-Wallis test are shown in figure 2.11. With 27 factors, one or possibly two would be expected to show up anyway as different at 5% level, so four is a surprisingly small number of differences. The differences in ratings shown in figure 2.11 are understandable in that common sense would indicate that specialists would regard their experience as more important than non-specialists.

One explanation for the broad level agreement may be a consequence of the survey design. Kaiser & Bostrom (1982) found that users in software design teams displayed the same personality characteristics as systems personnel. They suggest that this may be because people with systems-like personalities are selected or self-select for involvement in software development. This argument may apply to the respondents of this questionnaire, that is those choosing to answer the factors part have systems-like personalities.

Factor	% rating factor important			
	User	Manager	Specialist	Sig
Experience of the developers	79	80	91	.0018
Consideration of users' jobs	69	63	61	.0475
Experience in the organisation of this kind of development	53	55	70	.0401
Size of team	36	35	50	.0423

Figure 2.11 Significant differences between roles in rating factors

The following figures show significant differences detected for purpose of system 2.12, hardware 2.13, organisation (type 2.14 or size 2.15). The comparison between different system purposes suffers because the number of edutainment (10) and R&D systems (11) was much smaller than real-time (46) or information (166). There were no significant differences between the rating by male and female respondents. These tables show that it is only when the ratings are compared across system complexity that a number of significant differences emerge.

Factor	% rating factor important				Sig
	Edutainment	Real-time	Information	Systems /R&D	
Involvement of users	70	72	86	73	.0022
Computer resources available during development	56	75	69	100	.0169
Testing	44	87	80	90	.0425
Commitment of the users to the system	78	65	84	82	.0491

Figure 2.12 Significant differences between purpose of system in factor ratings

Factor	% rating factor important				Sig
	Stand-alone PC	PC network	Main-frame	Hybrid	
Monitoring and control	51	52	75	77	.0025
Introduction of the system	65	68	82	86	.0059
Testing	68	75	91	85	.0118
Realistic planning	74	66	81	81	.0182
External factors	15	7	26	35	.0266
Size of team	25	43	45	38	.0434

Figure 2.13 Significant differences between system complexity (using hardware type as an indicator) in factor ratings

Factor	% rating factor important							Sig
	Manuf.	Retail	Fin.	Educ.	Serv.	Local Govt.	Cent. Govt.	
Technical competence	87	61	96	64	91	100	89	.0118
Involvement of users	79	93	84	77	80	57	94	.0176

Figure 2.14 Significant differences between different types of organisation in factor ratings

Factor	% rating factor important				Sig
	<20	20-99	100-1000	>1000	
Involvement of users	95	74	81	80	.0223
Testing	70	66	78	91	.0308
External factors	12	28	28	20	.0421

Figure 2.15 Significant differences between different sizes of organisation in factor ratings

Overall these analyses reveal a surprising amount of agreement over the rating of factors both between different types of respondent and system indicating that a range of technological and human factors were important in all systems. There were more differences for system complexity than other system characteristics. The few differences significant at the 1% level were largely to be expected:

- specialists thought 'experience of developers' was more important than users or managers;
- 'involvement of users' was regarded as more important for information systems than other types of system;
- 'monitoring and control' and 'introduction of the system' became more important the more complex the system.

These findings support the decision to study software development as a whole rather than focus on type or complexity of system. Unfortunately the questionnaire did not allow the development environment to be distinguished.

2.1.6 Qualitative results of the survey

Respondents added comments about their system, about why individual factors were important or, more rarely, unimportant and added additional factors. In contrast with the quantitative results these comments are not representative; although 86 respondents made comments, the number of respondents commenting on each factor was only between 8 and 18, and so each point was only made by a very small proportion of the respondents. What is interesting, the comments illustrate the diversity of thinking behind the rating of the factors.

As the comments made in different parts of the questionnaire frequently reinforced similar points, they were drawn together for the purposes of analysis. The comments were analysed using NUD*IST and four broad categories emerged: organisation, project management, software tasks and technology. The main points that emerged from this analysis are summarised in the following paragraphs.

Organisational environment

Commitment was 'crucial' for some projects but often the commitment of individual managers was more important or more feasible than getting board level support. When respondents talked of the need for commitment, this was often in connection with change and the resulting 'doubt and uncertainty'. The alignment of IT developments with organisational strategy was not one of the listed factors but was added by several people and was connected with the issue of board level ownership of projects. Financial constraints of an organisation (and this may be the third party supplier of software) affected projects mainly through a restriction on the number of developers or the loss of 'good people'.

Project management

An organisation's lack of experience in managing large projects was one factor affecting successful project management. Lack of realistic planning and forced deadlines were also cited as factors. Some projects were made more difficult by physical distances or mixed cultures. Generally small teams were favoured 'to retain focus' even though it was recognised that this meant more elapsed time. Developers were expected to be self-motivated and new software meant that they had to learn as they developed. One respondent cited the case of a newcomer who altered the system for their own ends and then left the company. The traditional communication gap between developers and their customers was still cited as a problem.

Software tasks

When users were involved in defining requirements this sometimes caused problems because of the users' lack of technical ability, differences between the user and sponsor or difficulties in programming the design. Sometimes getting a project in on time was more important than usability. Prototyping was regarded as a useful technique for eliciting requirements and defining user interfaces. There were very few comments about other methodologies although PRINCE seemed to come in for both brickbats and bouquets. Quality procedures did not feature in the given factors but were alluded to by some. Testing and the lack of defects were regarded as important to system credibility. The

changes connected with the installation of systems, inadequate marketing and poor user training seemed to cause many implementation problems.

Technology

The technology itself caused problems for some projects; as one person commented: 'Users are after solutions not h/w or s/w. But these are critical to the company.' Problems were generally connected with not having sufficiently powerful equipment either because of under-investment or out of date equipment.

2.1.7 Summary

The survey was useful in indicating that there are a large number of factors that are important in software development. Although many of these factors have some technical component they are all people related and this was reinforced by the comments added by respondents. While this human emphasis on software development might have been expected from users and managers, section 2.1.5 showed that there were few significant differences between specialists, managers or users. The only real difference in factor rating was in system complexity. For this research, the survey justifies the need to look at human aspects of software development and shows that while the importance of factors may vary in detail (as indicated by the analysis of the comments), there is a surprising amount of similarity at a gross level.

However, the survey has also highlighted some of the inadequacies of this research approach. The pilot indicated that there would be problems of interpretation and there were a few cases where the comments contradicted the rating given to the factors. There is also the problem that the results of a survey inevitably reflect the questions that are asked or not asked. It is interesting to compare these results with a survey in 1989 of senior American MIS managers (Niederman, Brancheau & Wetherbe 1991). Despite the difference in participants, the analysis of section 2.1.5 would lead one to expect some similarity of results. However, there is remarkably little overlap; for example, the top three categories in their study are: developing an information architecture, making effective use of the data resource and improving IS strategic planning.

The comments also provided tantalising glimpses of project successes and failures but very little information. In order to explore more fully the factors and get behind the obvious problems such as ill-defined requirements, it is necessary to study developments in detail using more open-ended techniques.

2.2 LITERATURE REVIEW

Literature reviews can serve many purposes (Massey 1996 provides twelve analogies); this review provides ‘the scaffolding for building the justification’ for the research and acts as a ‘mirror’ reflecting my theoretical stance. The previous section showed that human influences are important in software development in a variety of complex ways, this section illustrates how software development literature treats human issues. In parallel the review establishes the background that I bring to the interpretation of the data described later in this thesis.

In reading this literature review it should be noted that it is not the traditional literature review of scientific inquiry. It does not provide the ‘building block’ (Massey 1996) or foundations on which this research is being built because within the research paradigm used for the rest of this research (see chapter 3 and particularly section 3.3.3) this is inappropriate. As a consequence it does not consider the extensive theory in the management literature (eg socio-technical, organisation, culture, power) which might be apposite in addressing the research question within a positivist paradigm.

One of the problems in undertaking this literature review is that ‘software development is a loose amalgam of different specialisms, not a single discipline in its own right’ (Jackson 1992 p33). Randall, Hughes & Shapiro (1993 p197) take the view that software development is inherently interdisciplinary because development is ‘inextricably bound up with use’. This research is intended to inform software developers and their managers so literature pertaining to software development has been reviewed from the disciplines of computer science, information systems and management.

An analysis was undertaken of the way in which human issues are handled by analysing 325 texts spanning the disciplines in NUD*IST. Many of the texts were chosen because they can be regarded as classics in their field. Each of the texts was categorised according to the way in which human issues are treated in the text. Analysis of this categorisation showed that the literature is polarised and presents five focused views of the human influences on software development.

Computer science tends to view software development from a technical perspective that is heavily influenced by programming. This has been reinforced by the engineering metaphor embodied in the use of CASE tools and formal methods. The real-time and object-oriented development methodologies within this view also generally have a focus on design and implementation rather than on the human problem that is being solved. Section 2.2.1 discusses the human issues in the software engineering view of software development.

In contrast, the information systems literature prefers to use the term systems development rather than software development to reflect the view that information systems are frequently developed to meet the needs of specific users within an organisational context (Jayaratna 1994). The view embodied by the systems development literature is discussed in section 2.2.2.

Within both computer science and information systems there is literature on the management of the software development process. Much of the literature has a technical rather than human focus although there are exceptions in both disciplines. In section 2.2.3 the human focus in the project management literature is discussed.

Two further views of software development emerge from an examination of the information systems literature. Of the ten subject areas which the UK Academy of Information Systems (UKAIS 1996) has identified as defining the information systems discipline, those concerned with the development and management of information systems (UKAIS areas 2, 3, 4 and 5) have already been covered. The organisational view of software development presented in section 2.2.4 summarises literature from information systems and management on organisational influences on software development (UKAIS areas 1, 6 and

7). This includes literature on organisational structure, culture, strategy, change and business process re-engineering.

The final section in the review, 2.2.5, discusses the social view of software development (UKAIS areas 8, 9 and 10). While much of this literature is about the influence that software has on society, there are some interesting insights on software developers.

In undertaking this review, it was disappointing to discover that despite the importance of software development, the process has no special treatment within the management literature. For example, Peters (1992) says that IT is 'causing the most significant change in the way we organise, live, make war, and do politics in a thousand years' but only devotes 36 pages (out of the 763 pages of 'Liberation Management') to the actual provision of information technology and much of this is about its use not development.

The following sections illustrate ways in which human issues are treated in the five views of software development. Examples of literature are chosen to characterise and contrast viewpoints, not to be exhaustive. The literature largely reflects my theoretical stance when I began the analysis of the case study work (in 1995); chapter 9 re-examines contemporary literature in the light of the findings.

2.2.1 Software engineering view of software development

Naur in his writings which represent forty years of trying to understand what computing is about (1992 pxiv) believes that a 'proper understanding [of computing] is predominantly a matter of human beings and their activity'. Weinberg's (1971) well-known and seminal work begins: 'This book has only one major purpose - to trigger the beginning of a new field of study: computer programming as a human activity, or, in short the psychology of computer programming' (Preface vii). Psychologists have studied behavioural aspects of software engineering since 1950s (Curtis 1984). However, the study of programmers as individuals has had little impact on mainstream software engineering literature. For example, Sommerville (1992 p22) has to justify the inclusion of a small chapter on human diversity, knowledge processing, group working and ergonomics in his technical book with

his 'conviction that an understanding of the people involved in software engineering as system users, specifiers, designers, programmers and managers helps with the technical processes of systems development'.

The focus of the study of individuals in software engineering has been on programmers and the programming task rather than on the managers of programmers even though it can be argued that these people may have an even greater impact on project performance (Curtis 1989). Weinberg (1971) identified that variations in personality, intelligence, training and experience of programmers affect their performance, by as much as an order of magnitude according to Brooks (1975). Performance can also be affected by characteristics of the job which impact the programmers' psychological state and hence outcomes (Curtis 1989).

One area of software engineering where the involvement of humans is recognised by some as being influential is design (Loomes 1991). Quintas (1995) identifies the tension between the creativity of design and the push to become a proper engineering discipline. This is exemplified by two well-known articles from the 1980s. Hoare (1984 p10) talks of 'a radical change in the development of large programming projects' based on a strict life-cycle and full formal specifications meaning 'there will be no bugs'. Around the same time, Parnas & Clements (1985) claim that the software design process can not be rational because of human influences although there are benefits to be derived from following methods as closely as possible and from documenting the design as if it had been produced rationally.

One of the problems that Parnas & Clements highlight is the difficulty of ascertaining requirements because the people who commission software do not know what they want and because they are affected by external change. The human problems of getting clear requirements had largely been ignored by the software engineering literature (in direct contrast to the systems development literature, see section 2.2.2) until recently when the term 'requirements engineering' started to be used. (1991 is the first year in the BIDS database where a search for 'requirements engineering' yields several articles.) Once again Weinberg is ahead of his time with Gause & Weinberg (1989 preface pxv, original italics): 'This book is about the *requirements process* - the part of development in which *people attempt to discover what is desired*.' Naur is unusual in recognising in 1971 that

programmers need to understand not only what but why. Nevertheless, the ‘multiplicity of ways in which social is incorporated into the requirements process is still very limited’ (Jirotko & Goguen 1994, p4).

This limited attention to the requirements process is reflected in software engineering methodologies that have developed from a programming/design focus. This is as true of the data-flow based real-time methodologies with their emphasis on control as the newer object-oriented methodologies. For example, Fayad et al (1993) conclude of the Shlaer-Mellor approach (Shlaer & Mellor 1988, 1992) that it is good for applications with clear requirements. Even newer approaches such as Fusion (Coleman et al 1994) begin by brainstorming an object model from the requirements - this is only possible if the requirements are fairly well understood. The introduction of the Unified Process (Jacobson, Booch & Rumbaugh 1999) has seen a significant increase in emphasis on requirements although ‘it is still not uncommon for project teams to start writing code (which is fairly easy) before they have firmed up just what the code is supposed to do (which is difficult)’ (ibid p112). This change has been influenced by the growth in the application of ‘Use Cases’, which focus much more directly on the role of users (Stevens & Pooley 1999).

The work on requirements engineering in software engineering has developed from the Human-Computer Interaction literature. HCI developed in the mid-1980s from an interest in the user interface to a wider concern with the design, evaluation and implementation of interactive systems (Preece et al 1994). The human focus in this work is largely upon the people using the systems rather than on the people developing those systems (Preece & Rombach 1993). While the early work concentrated upon the individual user and their interaction with a software product, the user’s organisational and social context is now considered to be relevant (in, for example, Preece et al 1994). The HCI literature also considers the involvement of the user in the process of software development. Initially, user-centred design approaches developed separately from the more user-focused systems development methodologies (see section 2.2.2) but Preece et al (1994) include brief descriptions of Soft Systems Methodology and Multiview. These issues in HCI are not well integrated with other software engineering literature and HCI is often taught as a separate subject.

In the software engineering view, programming and design may be seen as activities undertaken and influenced by human beings but there is little interest in the human aspects of the problem domain except for the work on user interface.

2.2.2 Systems development view of software development

Within the information systems literature, the problems of determining requirements have been discussed as much as the more technical concerns of implementation. Cougar's 1973 review of business systems analysis, charts the development of techniques which emerged in the 1950's. The emergence of the structured approaches to analysis has led to a dominance in the literature of a process view of software development in the information systems arena. This is epitomised by the importance of methodologies in the literature and in education. For example, whilst teaching on a computer science degree, heated debate on programming courses centred around which language to teach and on business systems courses which methodology to teach.

Methodologies are generally used directly by developers despite the predicted use of CASE tools and provide 'an explicit way of structuring one's thinking and actions' (Jayaratna 1994 p37). Given this it is not surprising that a specific methodology should reflect a particular view of reality and hence influence significantly the course of the development (Hirschheim & Klein 1989).

The classical structured approaches originating from the work of De Marco (1978) and Gane & Sarson (1979) which are now incorporated in the Yourdon (1989) and SSADM methodologies (eg Eva 1992) represent a strong orthodox approach to systems development (Hirschheim & Klein 1989). These methodologies 'tend to focus on well-established issues within computing science, perhaps because of a recognition of the complexity of analysing and then incorporating the social' (Jirotko & Goguen 1994 p4). Despite this the structured approaches arose out of a desire to solve some very human problems of software development. De Marco's (1978 p10) six major problems of analysis are: communications between people, changing nature of requirements, lack of tools, problems with the specification, work allocation within the development team and politics.

One of the failings of structured methodologies is that they are based on the assumption that the nature of the problem is well understood. This is an area addressed more generally by the soft systems methodology (SSM) (Checkland 1981, Checkland & Scholes 1990). While systems theory has influenced information systems development approaches (Avison & Fitzgerald 1988), the explicit use of SSM in systems development has been debated (eg Checkland 1992). The design of the early stages of the Multiview methodology (Avison & Wood-Harper 1990) was influenced by SSM. Multiview also incorporates the socio-technical approach advocated by Mumford (1979) in which a more active user participation in systems development is adopted.

The communication gap between developers and users is widely recognised (Quintas 1995). Jayaratna (1994) agrees that interpersonal relationships between clients and developers can have a significant impact on problem formulation and solutions recommendation but very few methodologies explicitly cover these issues. It has been argued that modern systems, which are not just mirrors of existing paper-based systems, require more proactive involvement of users (Kendall & Kendall 1994). In practice this seems to result in a user-centred or prototyping approach to development (eg Boar 1984), or more extensive user control of development through participatory design or end-user computing.

In user-centred and prototyping approaches there is a much closer interplay between problem and solution which is a formal recognition of the continuous process of design extending into the traditional maintenance phase (Quintas 1995) and the dangers of adopting a rigid life-cycle approach (McCracken & Jackson 1982, Gilb 1988). The Unified Process has adopted an iterative approach, like prototyping, but has none of the emphasis on user involvement.

Although the user is closer to the development in prototyping, the analyst is still 'running the show' in contrast to approaches where the user manages and possibly completes the development. Although reports of participatory design are encouraging, Clement & Van den Besselaar (1993) found that the approach had not become established even within the organisations where it was used. Conger (1992) predicts that end-user computing will continue to expand with the increased sophistication of personal computing technology even

though Quintas (1995) doubts the utility of much user-initiated programming because of the lack of cumulative experience in development.

Within the literature there are a variety of views about the adoption of methodologies. Amongst methodology supporters, one view is typified by De Marco & Lister (1987) who do not regard methodologies as a means of controlling developers and recommend standards of no more than 10 pages. At the other end of the spectrum is the software process maturity framework where Humphrey (1990) talks of the discipline of controlling the environment and methods by standards. There is a danger that methodologies, by providing a method, are assumed to be useful (Naur 1992) and consequently they should come with a health warning (Jayaratna 1994) that their usefulness depends on how they are applied (Loy 1993, Tagg 1991). Moreover, some feel that the concentration on methodologies is hampering progress on systems analysis (Benyon & Skidmore 1987). Despite the emphasis in the literature on methodologies their use in the UK is limited (Quintas 1995).

In the information systems view of software development, the people who will use the system are seen as central to the process and this is reflected in the ways in which users have been incorporated into the development process.

2.2.3 Project management view of software development

Despite a substantial literature on project management, which spans both software engineering and systems development, there have always been respected figures who do not believe management is important in software development. In a conference in 1969, Hoare said 'basically all problems are technical' (quoted by Weinberg 1971 p113) and over twenty years later, Meyer in the plenary for Object Technology '93 strongly denigrated the view that management is more important than technology. Despite this, there is a general acceptance of the importance of good project management: two years later, the plenary for Object Technology '95 given by Adele Goldberg centred around management issues and one of the software development principles extracted from writings of researchers and practitioners is 'Good management is more important than good technology' (Davis 1995 principle 127).

The differences in the literature on project management reflect alternative views on the nature of the tasks of software development. Glass, Vessey & Conger (1992) categorise the alternatives as 'clerical' (capable of being 'accomplished using routine procedures') and 'intellectual' (requiring 'non-routine thought processes'). Within the clerical view, the emphasis is on bringing order: 'A software manager has diverse roles but the most significant activities are project planning, estimating and scheduling.' (Sommerville 1989 p491). Even books such as Ince, Sharp & Woodman (1993) which have a rather broader definition of software project management, tend to focus upon what De Marco (1982) describes as 'the project's quantitative parameters' (ibid p1). This view ignores the fact that in practice project managers frequently have to ignore rational approaches to cope with conflicting stakeholder demands (Sauer 1993b). Weinberg (1982) thinks that this view dominates because software managers are promoted programmers so they use programming techniques when managing.

An alternative view is that it is inappropriate to see software development as a manufacturing process because of the intellectual nature of the tasks (De Marco & Lister 1987) and a more relevant analogy is of growing software (Quintas 1995). This view may result from recognition that software development is chaotic (Olson 1993) or that modern software problems require an unstructured process (Whitaker 1994). O'Connell (1994) is unusual in combining a structured approach to software management with helpful advice on some key human concerns.

One important consideration is the nature of software developers: 'Our industrial activities demonstrate that these needs [personal needs of problem solver] have an overwhelming influence on the designed solutions and on how methodologies are used in practice' (Jayaratna 1994 p68). However, there is a view that 'IS professionals are different from the general population of workers. They behave differently and hence should be managed differently' (Dengate, Cougar & Weber 1990). Motivating software professionals can be difficult because of the considerable autonomy exercised by developers (Quintas 1995) and the biggest motivator for a programmer can be being able to do the programming their way (Weinberg 1971). Whitaker (1994) illustrates the consequences of getting the reward systems wrong. There is also a limited literature on the effects of the working environment

of software engineers both in terms of the physical space and technical environment (eg De Marco & Lister 1987, Frangos 1998).

Teams are generally regarded as important in software development. Hyman (1993) describing the exponential growth of a software function over seven years saw team dynamics as the most important factor. Weinberg (1971) recognised the importance of studying software development teams and identified a lack of research but although O'Connell (1994 p35) recommends 'a small group of highly trained and focused people', there is relatively little written specifically about the characteristics of software development teams (Ford & McLaughlin 1992). Brooks' (1975) concept of the 'chief programmer' team seems to carry little weight amongst practitioners and was not even mentioned in an article reviewing the lessons learnt since his book was published (Verner, Overmyer & McCain 1999).

The implication of Belbin's work, particularly his comments on 'Apollo teams' (1981, 1993b) would appear to be highly apposite to software engineering. Belbin (1981 p15) cites one example from the software industry but little reference is made to Belbin's work in the software development literature. One exception is Henry & Steven's (1999) study, which uses Belbin's roles to investigate the importance of the leader role in software teams. Unfortunately this study is based on a very artificial task and takes no account of the other roles. Informal studies amongst computer science students and practising software engineers indicate an absence of people with preferences for adopting the 'chairman' and (to a lesser extent) the 'completer-finisher' roles. (The studies are limited by using the self-assessment questionnaire published in Belbin 1981 that excludes the 'specialist' role and are subject to the concerns expressed by Furnham et al (1993a, 1993b) and discussed by Belbin (1993a).)

Two of the six drivers for successful engineering teams identified by Thamhain & Wilemon (1987 p133) are concerned with 'experienced engineering management personnel' providing 'proper technical direction and leadership'. Despite this, there are mixed views in the literature about the need for technical expertise in software managers. Ould (1990), writing with the perspective of a technical director of a software company, stresses the

importance of understanding the problem in successful project management, while Ives & Olson (1991) found that the role of senior information systems managers is primarily managerial. Focusing on complex, uncertain technical issues can mean that interpersonal matters are neglected (Zachary & Krone 1984). Even though technical proficiency is not a leadership trait, it can be important for gaining the confidence of developers (Whitaker 1994).

Even within the planning approach to software project management, human issues are important. The point of planning is to manage risk and the unknown (Ould 1990) and much of the unknown turns out to be human-related; for example the productivity of individual engineers (Ould 1990) or the effect of the estimator on the accuracy of estimates (De Marco & Lister 1987). One view is that the real problem of software development is of communication between people and that the techniques of project management build 'better communication bridges' (Gilb 1988 p23).

Software project management literature assumes the importance of the team and although there is limited research on effective software development teams some literature recognises the importance of viewing the developers as individuals.

2.2.4 Organisational view of software development

Efforts in the 1980s to make software development into an engineering discipline went into tools and techniques, not organisational aspects (Quintas 1995), even though most computer-based information systems are developed and used within the context of an organisation (Walsham 1993). Despite this, most information systems methodologies are not concerned with what really happens in organisations (Jayaratna 1994). The effects of organisation structure and culture on the development of successful information systems are rarely considered, although as Heiskanen (1993) shows, these can be very important. A few studies have shown the importance of understanding the culture in the developing organisation (eg Fischer et al 1993, Dubé & Robey 1999). Walsham (1991) drawing on the work of Morgan (1986) on organisational metaphors stresses the importance of drawing on organisation theory in information systems research.

In contrast, during the last 20 years the literature on information systems planning has developed from an emphasis on technical issues to a focus that incorporates business considerations (Galliers 1995). Nevertheless, the outcome of strategic IS/IT planning is still largely seen to be about identifying the most appropriate targets for automation and scheduling that automation (Fidler & Rogerson 1996, Robson 1994, Baets 1992). Although IS/IT strategy planning is popular (both of the large organisations studied for this research completed some form of strategy study), limited success with the results are reported (Flynn & Goleniewska 1993, Ward, Griffiths & Whitmore 1990).

One of the problems for business-aligned IS/IT strategic planning is that strategy plans have become more short-term because of the rapidity of change both in technological developments and organisational environments. In Tagg (1994b), I argue that there is a need to learn from writers on management who talk about learning to cope with instability (eg Stacey 1992, Peters 1992 and Morgan 1993). In the context of business strategy, Mintzberg (1994) distinguishes between visions and plans and Eardley, Avison & Powell (1995) support my view that vision is required in information systems strategy. Unfortunately, few management writers on strategy consider IS/IT strategy explicitly.

Bennett & Hinton (1994) identify an over-emphasis on hard approaches as being a cause of failure in IS/IT strategic planning. Closer alignment of IS/IT with organisational objectives means a change in role for systems analysts with a greater emphasis on human relation skills (Burgess & Hughes 1996). Moreton (1995) identifies these skills rather more precisely to be: collaborative working in interdisciplinary teams, use of socio-technical design principles, change management and the development of adaptive systems. However, much of the information systems literature has concentrated on the content of change rather than the process of change (Walsham 1993).

While there is literature on how information systems change organisations (not the emphasis of this research), there is less interest in how changes within the organisation affect software development. In business process re-engineering (BPR) literature there is a tension about the role of software development. Hammer & Champy (1995) see information technology as an important enabler of BPR. Relegating information technology to a support

role in organisational change means ignoring the process innovation that can come from IT (Edwards & Peppard 1994).

Although software development is recognised as an important element in BPR (Fiedler, Grover & Teng 1994) and is regarded by some (eg Earl 1994) as responsible for the early growth in BPR, the literature largely ignores how the integration of BPR and information systems development may be achieved (Demay & King 1996, Weerakkody, Bennett & Tagg 1996). Although Quintas (1995) sees that BPR places developers at the centre of organisational development, in practice there is often an organisational division between the BPR and IS/IT development teams (Weerakkody, Tagg & Bennett 1995).

There is a need to move away from the traditional approach to software development based upon work-study principles to a greater concern with the organisational context (Moreton 1995). However, one of the problems for software development when technology drives BPR is that requirements are by necessity ill-defined which is problematic for technically-oriented IT staff (Willcocks & Smith 1995) and traditional development approaches (Eardley, Avison & Powell 1995). It has also been argued there is a danger of marginalising human and political processes which are key to successful BPR (Willcocks & Smith 1995).

The organisational view of software development is limited by the relatively little integration of management and information systems literature on organisational influences on software development. There is further management literature that appears to be relevant and applicable to software development.

2.2.5 Social view of software development

The emphasis in the previous section was on the organisational influences on software development, this section considers the literature that takes a wider view of the interaction between computers and society as a whole. Information specialists have a responsibility to their organisation and society at large (Jayaratna 1994) as illustrated by the literature on

ethics including the effects of technology on working practices. This literature has now broadened out to include the effects of technology on all aspects of life (eg Kling 1996).

There is also a growing but somewhat diffuse literature that puts the social side of computing first rather than seeing it 'relegated to a kind of explanatory storehouse' (Bloomfield & Vurdubakis 1994a p4). They take the view (1994b) that for too long technical systems have been seen to have human consequences but the human influences on the development of technical systems have not been considered. Within this literature there is a strong call to question the boundary between the social and technical and in many cases this is accompanied by alternative research approaches influenced by social science.

The influence of social science is both useful and problematic in this area. The application of critical theory and constructivist sociological perspectives described by Murray & Woolgar (1991) provides a means of broadening the 'conceptual horizon of inquiry' (Bloomfield & Vurdubakis 1994a p4). This is the rationale for adopting a non-positivist research approach in this thesis (see chapter 3). However, the influence of social science means that literature such as the emancipatory framework used by Kendall & Avison (1993) to categorise the work from the IFIP WG8.2 working group or the emotive language of Star (1995), can be inaccessible to practitioners in software development. There have also been difficulties reconciling the objectives of sociologists and technologists (Low et al 1996). While technologists are looking for action to improve the process of software development, sociologists may seem to be just giving an account of what they observe.

Despite these problems work in this area has shed some useful insights on the processes of development. For example, the study by Button & Sharrock (1994) of the use of a new methodology, CASE tool and programming language to develop photocopier software, shows how the use of these tools depended on the social and organisational circumstances. In consequence, they argue that the utility of software development tools should be considered in the context of their use. Murray & Willmott (1993) used a different research method to investigate a user-centred approach to development. This shed some insights on the difficulties of communication between developers and users.

One of the consequences of blurring the distinctions between social and technical is that the technical is inadequately addressed (as in Low & Woolgar 1993), which may arise from the relative infancy of this work (Star 1995), or from a lack of technical understanding deriving from a sociological rather than computing focus. It is hoped that this thesis will help to bridge the social/technical divide identified by Murray & Woolgar (1991 p4).

The social view of software development sees all social issues as important but does not present a coherent view because it is a relatively new field. The interdisciplinary nature of the literature means that sometimes the technical is inadequately addressed or the results are inaccessible to a technical audience.

2.3 DISCUSSION

The survey reported in section 2.1 showed that respondents considered a range of factors, both social and technical, important for success in software development. There were remarkably few significant differences in opinion whether the sample was stratified by type of respondent, type or complexity of system, or type or size of organisation. However, the survey also illustrated the limitations of this type of research. Responses were influenced by the list of factors in the questionnaire which appear to have been misinterpreted in some cases. Moreover the open-ended comments indicate the importance of factors but do little to explain their significance. Chapter three explores alternative research approaches.

The literature review illustrated how human issues are handled in the software development literature. Most of the relevant literature came from the disciplines of computer science and information systems. There was surprisingly little explicit reference to the process of software development in management literature. The handling of human issues was characterised by five views of software development:

- software engineering
- systems development
- project management
- organisational

- social

Each of these views recognises the importance of social factors in software development but the literature is fragmented. For example, an issue such as the impact of the skills and personality of the developer as an individual has been recognised for many years within the software engineering view, but has largely been ignored in the systems development view and only partially considered in the project management view. This illustrates the importance of taking a holistic approach to the study of software development as a human activity.

There is a growing awareness of the importance of considering the human influences on software development which is most marked in the literature characterised by the social view although it appears in other areas. In the special issue of Communications of the ACM on Participatory Design, the editors ask: ‘must we always analyze the impact of *technology on people*, or is there just as strong an impact of *people on technology*?’ (Kuhn & Muller 1993 p26, original italics). Unfortunately there is a lack of maturity in the social view of software development exemplified by lack of integration with more technical considerations. For this reason this research will explicitly consider technical dimensions when exploring software development as a human activity.

The research into social influences on software development described in section 2.2.5 is ‘using a variety of research approaches. This provides further justification for the investigation into an appropriate research paradigm described in chapter 3.

3 Research paradigm

The previous chapter established that the research will focus on software development as a human activity and indicated that a survey approach to the research was likely to be inappropriate.

Human issues in software development are included in literature from computer science, information systems and to a lesser extent management disciplines. So it seems appropriate to begin the search for a research approach in the traditions of those disciplines. Section 3.1 reviews the research approaches adopted in each of these disciplines. Although research method is sometimes not made explicit, particularly in computer science, in all three disciplines the dominant approach is underpinned by positivist beliefs. Qualitative approaches are used in information systems and management but there is not an established body of literature covering both techniques and the underlying inquiry paradigm.

During the progress of this research it became increasingly less tenable to justify the positivist view of a single objective reality with an emphasis on theory. The journey I undertook to arrive at an alternative underlying inquiry paradigm for this research and how it resulted in the adoption of a constructivist-based paradigm is described in section 3.2.

The research paradigm is discussed in section 3.3 and the basic framework for the research is established. The implications of the research paradigm on research quality are discussed in section 3.4 and criteria for quality are established.

3.1 APPROACHES IN CONTRIBUTING DISCIPLINES

As already discussed in chapter 2, this research spans the disciplines of computer science, information systems and management. This section reviews the research approaches prevalent in each of these disciplines at the time this review was undertaken. Developments in research method since then are discussed in chapter 9.

3.1.1 Computer science research

While Wegner (1976) identifies a trend from empirical scientist, through mathematician to engineer in computer science research, Fitzgerald et al (1985) merely report that the scientific approach is endemic. However, the underlying theories, necessary for the systematic scientific study of software engineering process, have not yet been established (Weide & Defazio 1993). Zave's (1991) retrospective description of the PAISLey project is typical of research in this area. Her review of nine years work spent on the development of an executable specification language reflects on the problems for research method when the project starts with 'ideas for a language rather than with a precise definition of a target class of system' (p214).

Wegner (1976) is unusual in discussing research method, the view implied by Ross & Staples (1994) is much more common. In describing the innovations at Southampton Institute in response to the government white paper on research methods courses, they describe sessions on such things as communication skills, writing, or library searches. This lack of awareness of research method was confirmed in conversations with computer science researchers at the University of Hertfordshire and the Open University (email communications with Ince (1993) and others) and by examination of PhDs in computer science (eg Baille 1992, Scholefield 1992). Loomes (1991) working in the area of curriculum design in computer science does discuss research method and implies a scientific norm for research in computer science.

There are a limited number of exceptions, which are using naturalistic approaches in computer science research. Hovenden et al (1994) illustrate the usefulness of ethnography in understanding software quality. This is part of a small but growing movement to use ethnography in computing although in most published work the ethnography is being used 'for and within IS development' (Beynon-Davies 1997) particularly in the area of user interface to gain a greater understanding of the working environment (eg Hughes et al 1993). Dick & Rouse (1994, 1995) illustrate the use of grounded theory to investigate the adoption of object-orientation.

3.1.2 Information systems research

Information systems is much more of an applied science than computer science because of the explicit concern for the organisations within which information systems are used. This leads to a greater awareness of social science and hence more concern with method (Work 1994). This is exemplified by comparing the difference between the approach to research methods described above for computer science doctoral students with that adopted by the University of Salford (Wood-Harper, Miles & Booth 1993). Their doctoral programme in information systems includes seminar-based sessions on foundations of inquiry and research methods.

Despite a much greater range of publications covering research method, Jenkins reported in 1985 that 'many MIS faculty and most MIS doctoral students are simply not research literate' (ibid p114). Attendance at the annual PhD consortium of Information Systems in 1993 indicated that this was still true for doctoral students from some departments, probably because of the difference in attitude to research method between Software Engineering and IS departments noted by Galliers (1993b). This lack of awareness of research method is reflected in publications with Davis (1992) reporting from a review of journal articles that there is little academic research in systems analysis and design. At a conference in 1988 that aimed to assess the state of research in systems analysis and design, many of the papers did not mention research method explicitly (Cotterman & Senn 1992). Cotterman & Senn concluded in their summary that 'the question of appropriate research tools and strategies generates more heat than light' (ibid 1992 p459).

The interdisciplinary nature of information systems means that the need for a plurality of methods is recognised (Fitzgerald et al 1985). Hirschheim (1985) identifies the need to draw on the social sciences for an epistemology while Antill (1985) sees information systems as a hybrid activity involving technical, personal, organisational, and philosophical aspects and thus drawing on research methods from different disciplines. However, as Avison & Myers (1995) point out, the choice of research method often depends on more practical factors such as the predilections of the funding body or academic department of the researchers.

Despite this plurality, the underlying paradigm is positivist in most cases. Jenkins (1985) describes 13 different research methodologies (based on his experience as editor and running PhD seminars) but he exposes his scientific assumptions when he orders his methods on the strength of hypothesis testing and states 'variables are, of course, central to all research' (ibid p112). Hirschheim's (1985) description of the development of an IS epistemology is largely a history of positivism and when Iivari (1991) analysed the epistemology of seven schools of Information Systems development by examining textbooks he found it to be almost entirely positivist. In many cases this outlook is not even questioned (Orlikowski & Baroudi 1991). One problem highlighted by Klein & Welke (1982) is that information systems can only be regarded as a scientific discipline because of a community of shared beliefs and little scientific progress has been made because of the lack of theory (Farhoomand 1992).

Nevertheless, most information systems research is either theoretically grounded (implicitly or explicitly) or descriptive (generally case-study based) (Orlikowski & Baroudi 1991). Jenkins (1985) describes a 'fundamental research process' (p103), which is presented as a linear process with feedback loops that begins with library research. More recently, it has been accepted that research approaches using case study or action research provide some grounding in data for theory building (Galliers 1992), and Gable (1994) suggests that case studies may be used to provide hypotheses to be tested through survey research.

The practical emphasis in the discipline (Avison & Myers 1995) and an understanding of the uniqueness of each system (Vitalari 1985) probably account for the extensive use of case studies and action research in information systems research. Multiview (Avison & Wood-Harper 1990) and Soft Systems Methodology (Checkland & Scholes 1990) are both well-known examples of the outcomes of action research. However, the concerns with the usefulness of single case or action research (Cotterman & Senn 1992) may arise from a lack of detail (dare one say rigour) in the way in which the cases are used in theory-building. 'Without such a framework action research can quickly become indistinguishable from mere action' (Checkland 1991). For example, Wood-Harper (1985) in a description of the development of Multiview does not explain how the action research was conducted or how the theory developed. In general, there is a shortage of material describing exactly how qualitative research in information systems is conducted (Rouse & Dick 1994). Walsham (1993) argues that the use of case studies needs to be related to the epistemology of the researcher.

Although Orlikowski & Baroudi (1991) only found 5 journal articles (out of 155 surveyed) to be interpretive in outlook, a variety of interpretive approaches are used in information systems research. Antill (1985) identified the following tensions in information systems research: repeatability of results vs. uniqueness of an information system, impartiality of the observer vs. experience of the designer, analysis of data vs. hypothesis testing, all of which explain the interest in interpretive approaches. Grounded theory has been used to study tool adoption in information systems development (Land, Le Quesne & Wijegunaratne 1992, Calloway & Ariav 1991) in a similar way to that used by Dick & Rouse (1994) to study object-oriented adoption. One virtue of grounded theory is that it 'provides a framework for the qualitative elicitation of the actors' view of a complex situation' (Little 1993 p229) and so may be an attractive option for researchers with little experience of qualitative research. Low & Woolgar (1993) use ethnography, Boland & Day (1989) phenomenology, Button & Sharrock (1994) ethnomethodology and Hopkins & Spaul (1994) recommend that critical thinking should be applied to information systems teaching. Interpretive approaches are regarded as most useful for research concerned with IT in organisations rather than more technological questions (Galliers 1993a).

Although interpretive approaches have much to offer information systems research, there are problems in borrowing methods from other disciplines in that researchers may not be aware of the nuances in the literature (Avison & Myers 1995). In part this is due to the absence of a clearly defined inquiry paradigm. This is illustrated by Galliers (1992) who categorises various information systems approaches that are mostly research methods with an underlying presumption of ontology and epistemology. In some cases this presumption is clear cut as in laboratory experiments but in others, such as case studies, it is not.

3.1.3 Management research

In management research, interest in, and concern for, research method is well established. For example, Bennett (1982) provides an extensive bibliographic guide covering philosophical issues as well as methods and techniques. Nevertheless, awareness of research method is not universal. For example the book 'Current Research in Management' (Hammond 1985) from a conference of teachers in management makes very little reference to research method and neither did many of the papers for the British Academy of Management Conference in 1993 despite research being the theme. The rapid expansion of business schools with an emphasis on teaching has led some universities to neglect research (Commission on Management Research 1994).

Researchers may not feel they have to define and defend their method because the prevalence of the positivist tradition means it is simply how research is done (Cassell & Symon 1994). Qualitative methods may also be used less often in management research because of ignorance in the academic and business community (Gummesson 1991) or pressures from industrial sponsors (Lowson & Rich 1995). Cassell & Symon (1994) go further and argue that qualitative research may be less valued due to the 'power of quantification in our western culture' which leads to a belief that somehow numbers represent the truth. This positivist tradition is also underlined in most management education (Holloway, Skinner & Tagg 1995 now published as Skinner, Tagg & Holloway 2000).

Despite the dominance of positivism in management research (Whitley 1984), there is an awareness of other views. For example, Easterby-Smith, Thorpe & Lowe (1991) contrast positivism with phenomenology and declare themselves to be social constructionists. Gill & Johnson (1991) starting from the perspective of the practising manager also stress the positivist approach while recognising that this is not the only approach.

Burrell & Morgan (1979) present four paradigms based on a subjective-objective axis and a regulation-radical change axis. The paradigm most associated with positivism (objective/regulation) they label 'functionalist' and classify phenomenological approaches (subjective/regulation) as 'interpretive'. With each of these paradigms they associate a view on ontology, epistemology, human nature and methodology. Thus they see quantitative methods being associated with the functionalist paradigm and qualitative with the interpretive paradigm. Bryman (1989) argues that it is not sound to link method with epistemology in this way and this difficulty highlights one of the problems of understanding management research literature. Some authors imply an epistemology by using the terms qualitative and quantitative, while others are merely using method within a positivist epistemology.

A number of good reasons are advanced for alternative research approaches. Richard Hackman in the preface to Van Maanen, Dabbs & Faulkner (1982) refers to the restrictiveness and narrow focus of discipline-based research in the study of organisations. This point is also made by Cassell & Symon (1994) who argue that a holistic view is more appropriate for research focusing on organisational processes that are subject to complexity and change. They also stress the impact of the individual situation on the research process, which argues for a focus on case study research (Gummesson 1991). Eisenhardt (1989) presents a convincing description of how case study research can be used to build theory, using ideas developed from grounded theory, within a positivist paradigm. Easterby-Smith, Thorpe & Lowe (1991) highlight the problem of researcher independence and see action research as a way of making a virtue out of this.

For these reasons and because of a disenchantment with quantitative methods, there is a growing interest in qualitative methods from social sciences (Van Maanen, Dabbs &

Faulkner 1982) but it seems that qualitative approaches are under-reported despite their widespread use (di Gregorio 1995). This is reinforced by the qualitative literature, which hardly mentions management research; for example, Tesch (1990) has separate chapters on Education, Sociology and Psychology but nothing on Management. This lack of a literature together with the uniqueness of each researcher's approach (Van Maanen, Dabbs & Faulkner 1982) makes it difficult for new researchers to take anything but a positivist approach.

3.1.4 Summary of approaches in contributing disciplines

In summary, the three disciplines contributing to this research are largely positivist in tradition. Whereas within computer science this is not questioned, there is more awareness of alternative approaches in information systems and management. However, in both information systems and management, within the debate on the plurality of research methods, there is a tension between relevance and rigour (Keen 1991, Commission on Management Research 1994). Nevertheless, even within management, where qualitative approaches have been used and published for over a decade, there is not an established body of literature to refer to which covers techniques and also the underlying ontology, epistemology and methodology of the inquiry paradigm.

3.2 DISCOVERING AN APPROACH

The overwhelming influence at the start of this research was consequently positivist, both in terms of the contributing disciplines and my own background. My first degree is in Mathematics and second in Computer Science. The dissertation for my MSc project (Tagg 1979) is very much in the mould of the computer science PhDs cited earlier. Initially I saw research method as unimportant, uninteresting or irrelevant and although I read many of the sources cited in the previous section, I did not make sense of what I was reading. In this section, I summarise the journey I undertook to arrive two years later at a research approach that is based on constructivism and uses qualitative methods. A fuller detail of this journey is contained in Tagg (1996a) and Open University (1998).

I have chosen to include this explanation because of the personal nature of qualitative research which 'is inexorably intuitive and implicit - internal and integral to the human being as a researcher' (Meloy 1994 p7). Moreover, as a researcher I come from a particular community which affects my interpretation of events (Denzin & Lincoln 1994). It is written in the first person to reflect the nature of the material (Meloy 1994).

My background is of a practising systems developer. I worked for ten years for the Bank of England based first in a user department and then in the professional systems development department. Following a secondment to the London School of Economics to work as the software developer on a research project (Cook et al 1980), I developed systems to support the Bank's internal financial functions. I used a radical (at that time) prototyping approach based on the early paper by McCracken & Jackson (1982) which enabled me to be much closer to my users than was normal. In 1985, I left the Bank of England to lecture in computer science at Hatfield Polytechnic (now the University of Hertfordshire). Whilst at the University I continued to develop systems on a consultancy basis and it was a mismatch between the practice of these consultancy projects and the theory of my lectures in systems development that provided an impetus for this research work (see chapter 1).

3.2.1 Theory informed by data

In the first year of research, my concerns were mainly with theory and the difficulty of handling diverse amounts of theory. I seemed to suffer from Gummesson's chinese box syndrome: every time I opened a box I found myself in a bigger box (1991 p18). One of the problems was the breadth of my work as I had already decided that I would undertake a research programme acceptable from both a management and a computing perspective despite the differences between PhDs in the disciplines (ABRC 1993).

In undertaking a management PhD, I needed to demonstrate an understanding of research methods (Gill & Johnson 1991). I found the arguments about research philosophy a distracting side issue and agreed with Hughes (1990) that there is 'no necessary reason why the solution of philosophical problems should be a precondition for social scientific

research' (p160). However, I was unconvinced by the rather simplistic approaches which are depicted as a series of steps moving in an orderly progression through data collection and analysis (eg Gill & Johnson 1991, Evans 1968). The naiveté of these approaches reminded me of the waterfall approach to software development. I was much more convinced by the erratic and iterative nature of research portrayed by Feyerabend (1975) and Frost & Stablein (1992) who describe research as an 'imprecise, creative craft' that is 'subject to serendipity as well as to plans' (p290).

Within the first week of the research, I had recognised the importance of following software development through time. My own experience indicated that the impact of decisions and factors influencing systems development was often only apparent after the system had been implemented (maybe years later). The value of longitudinal studies in research in information systems is recognised (Vitalari 1985, Galliers 1985, Davis 1992) but they are seldom used (Orlikowski & Baroudi 1991) possibly because of difficulties in obtaining long-term repeated access and disincentives in academic promotion (Davis 1992).

In consequence, I proposed to refine scientific knowledge using experience from the study of a small number of rich longitudinal cases in a feedback loop approach (Sparkes 1981) combining both theoretic approaches and hypothesis testing (see figure 3.1). This approach parallels experiential learning theory which puts knowing by apprehension and comprehension on an equal basis so that knowledge emerges from a dialectic relationship of the two (Kolb 1984). It was also in line with the ideas expressed in the 1984 IFIP WG 8.2 colloquium on research methods in information systems where Vitalari (1985) talked of needing a research method which permits learning and feedback into the method and Wood-Harper (1985) stressed the feedback elements in action research.

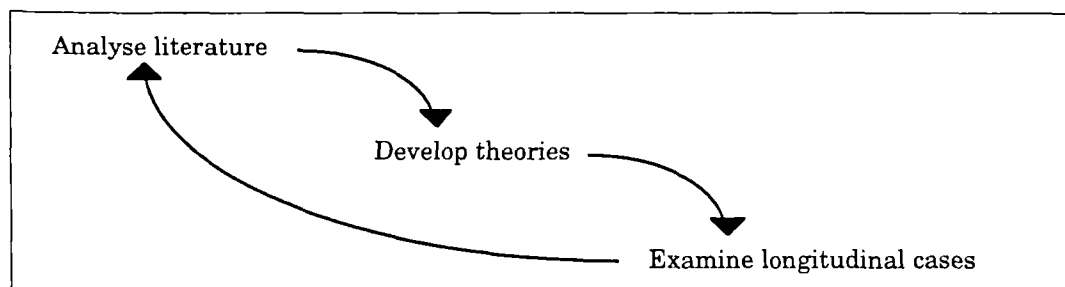


Figure 3.1 Feedback approach adopted during first year of research

The research outcome was to be a framework of the influences on software development derived from the literature and informed by the longitudinal studies. To assist with the process of synthesising the relevant but diverse literatures into a framework, I turned to mind-mapping. Although much of the literature on mind-mapping (Buzan 1989, Svantessan 1989) applies to the representation of received information, mind maps can be used for the innovative structuring of information that is part of research. In this use the map is continually evolving and it is necessary to have notes and citations for the branches so it is especially useful to use a computer. Existing mind mapping tools focused on diagrammatic aspects so I built a prototype writing tool for classifying theory based on the mind map concept (Tagg 1993)¹ using Hypercard².

During this period I was beginning to think about the longitudinal cases that were to inform the theory in the feedback loop. In the first case, V (see chapter 1), I was mainly concerned with supporting the project manager who had no experience of developing software systems. To pilot the approach to be taken in the other longitudinal cases, I became involved with a very small manufacturing business, W, who were considering buying their first computer. The action-oriented approach is still very evident in that I proposed to use a number of approaches to see which was most effective at identifying 'why' W needed a computer. As part of this I analysed W using Process Quality Management (PQM) (Ward 1990) writing up the work in a traditional passive style (Tagg 1994a). In discussing the case it was noticeable that important information had been lost in the focus on PQM and the writing up process. In consequence, in the remaining three organisations, I took a more passive role engaging in open-ended, taped, conversations with the main players and collecting development documentation. In this early work there was also an implicit focus on information systems which I sought to counter by studying two cases at Y, a network R&D company.

¹ Subsequently several commercial software mind mapping packages have been released which include this capability (<<http://www3.taggoram.co.uk/mapping/Mapping%20Packages>>)

² Hypercard was one of the first commercial implementations of hypertext and was included with early purchases of Apple Mac computers (Smith & Weiss 1988).

3.2.2 Qualitative quagmire

After six months of serious data collection with six longitudinal cases in five organisations under way, I began to become overwhelmed with the data. At this point, I discovered qualitative data analysis through Hypersoft (Dey 1993) and my continued interest in Hypercard. This was attractive because it appeared to retain the kind of rigour needed for 'good' science (Strauss & Corbin 1990 p57) whilst coping with the quantity of data I had already amassed. This was the beginning of a year spent trying to make sense of the literature on qualitative research which is 'not unlike a swamp populated with a variety of exotic beasts – "isms" and "post-isms" abound' (Phillips 1992 page ix).

It soon became clear that qualitative research is more than the analysis of non-numeric data but means such different things to different people that it is hard to codify (Tesch 1990). At the simplest level there is an awareness of the characteristics of qualitative descriptions which must take account of context, intentions, actions and consequences (Dey 1993). Despite the variety and individuality which characterises qualitative research, Van Maanen in the introduction to Van Maanen, Dabbs & Faulkner (1982) identifies five general principles: detailed observation as primary data source, observation of actual events, study of ordinary behaviour, consideration of human meanings and intentions, and, a descriptive focus. Miles & Huberman (1994) include these principles in their list of core recurring features (pp5-7) but add: the aim of gaining a holistic overview, the possibility of many interpretations of the data and the role of the researcher as research instrument.

Moving away from these concerns of data collection and analysis to more general issues of ontology and epistemology, the literature becomes difficult and confused (Harre 1981) particularly for someone from a positivist tradition as mirrored in Gardiner (1993) and in ongoing discussions on the QUALRS-L³ electronic list (eg Chamberlain 1995, Shank 1995). The stereotyping and polarisation in the literature is exemplified by Easterby-Smith,

³QUALRS-L, Qualitative Research for the Human Sciences, is an email list based in the USA with about 1000 subscribers worldwide. To subscribe send a message to LISTSERV@LISTSERV.UGA.EDU

Thorpe & Lowe (1991 p22): 'In the red corner is phenomenology; in the blue corner is positivism.' A variety of terms (such as naturalistic, interpretive, phenomenological or just qualitative) are used to refer to non-positivist research but in some contexts imply rather more specific world-views or methods. Another cause for confusion is the limited boundaries of much literature so that a text from educational research will place a different interpretation on terms than one from sociology (Avison & Myers 1995).

3.2.3 Discoveries

Given the research traditions in the contributing disciplines of computer science, information systems and management and my background it is hardly surprising that this thesis was begun within an implicit positivist tradition. During the journey of discovery described above, three important interrelated strands emerged which have shaped the approach adopted in this research. These may be loosely characterised as ontological, epistemological and methodological, which define the inquiry paradigm (Guba & Lincoln 1994).

My concern for method or how the research was to be conducted was gradually overlaid by awareness that this can not be separated from underlying ontological and epistemological beliefs. So, for example, when using case studies in research, the way in which the data is to be used to generate or support theory affects the choice of case study and the way in which it is conducted. This is also affected by the stance that is being taken on objectivity.

The next section builds on these personal discoveries to define the research stance that has been adopted for the research.

3.3 RESEARCH STANCE

The following discoveries of the previous section seemed to indicate that some form of qualitative approach should be used:

- the desire to take a holistic perspective on the problem;

- the idea that theory should be generated from data;
- the difficulty of justifying the existence of an objective reality when palpably users and software people see things differently;
- the fragmentary way in which human influences on software development are treated in the literature;
- the unique and individual nature of each software development that makes experimental approaches of limited usefulness.

Having decided that a qualitative approach is appropriate for this research, the obvious (scientific) solution is to adopt a well-defined, existing qualitative approach that fits the nature of the research question. Unfortunately, the existence of separate well-defined research approaches is illusory. Although there appears to be some consensus over ontological and epistemological concerns, there is a separate non-overlapping consensus over methodological issues. Moreover, as Tesch (1990) illustrates, the same term can mean different things within different disciplines. Within the disciplines contributing to this research there is very little literature that establishes coherent, complete research paradigms in the qualitative tradition.

In consequence, and justified by Tesch's (1990) principle that each qualitative analyst has to find their own way, the research stance is described in this section by giving my response to the three questions on ontology, epistemology and methodology, which according to Guba & Lincoln (1994 p105) define the 'basic belief system or world view that guides the investigator':

3.3.1 Ontology

'What is the form and nature of reality and, therefore what is there that can be known about it?' (Guba & Lincoln 1994 p108)

According to the original positivist definitions, reality is what is available to the senses (Hughes 1990). Post-positivists contest this view on two grounds. Firstly, theoretical objects which cannot be 'verified in terms of sense experience' are acceptable (Phillips 1990 p39). Secondly, the background knowledge of the observer affects what the observer sees so that 'we approach everything in the light of a preconceived theory' (Popper 1970 p52).

However, while post-positivists agree that people may have different, possibly conflicting

beliefs about reality and that there may exist competing theories to explain reality, there is nonetheless at most one view that is right (but which view may not be known) (Phillips 1990). In contrast, relativists believe 'that there is no single tangible, fragmentable reality' (Lincoln 1990 p77) because reality is a social construction.

The development of information systems is embedded in social context (Orlikowski & Baroudi 1991, Vitalari 1985) so the relativist approach would seem to apply. However, Phillips (1990) rejects the implication that seeing knowledge as a social construction implies multiple realities. He argues that relativists are confusing what people believe to be true with what really is true.

Consider a concrete, but simplified example. There is a computer program that the programmers regard as working and the users as not working. Who is right? Of course, it depends upon what you mean by 'working'. We could redefine this to be 'performing according to the specification' and then if we write a careful unambiguous formal specification it is possible to define 'working' so that one reality exists. (Actually, for a non-trivial program, we would not know that it was working because of the difficulties of proving correctness.) This argument implies that, if we define our terms carefully enough, we can arrive at a single reality. The trouble is that within a formal specification it would probably not be possible for the users to define all the characteristics they would want in a 'working' program. For example, the users would probably like all people who use the program to be able to get it to perform to specification reasonably easily. As we try and quantify this statement in order to make it precise and useful as an objective test of workability, it becomes less useful. A simpler solution is to fall back on the idea of multiple realities in which case there is the reality of the programmers within which the program works and a different reality for users in which the program does not work. This is similar to the examples given by Bloomfield & Vurdubakis (1994b) where the acceptance of a hospital information system depends on the appropriate people being convinced. In practice there may be a variety of user views dependent on their relationship to the system.

Taking the stance of multiple realities changes the nature of the research endeavour. Instead of trying to arrive at a definition of correctness that all parties can agree to and is

well enough defined to be useful, we can separately define what is meant by correctness in the multiple realities and explore the connections (Orlikowski & Baroudi 1991) and contradictions within the definitions. This will at least help both users and programmers to enrich their own definitions.

In the above discussion, in contrast to Schwandt's (1994) view that there is a tension between the idea that knowledge is individualistic and the idea that knowledge may be shared, I have assumed the existence of shared, publicly understood realities. This seems to me to be a perfectly plausible consequence of knowledge being a social construction. It fits the data in that people do seem to be able to discuss the world as if they shared a reality even when they may not share a common language. This was apparent during my conversations with programmers at Y where at times they could have been using a foreign language and yet, because of our shared experience as programmers, I could usefully discuss their work. This is in contrast to Low's experience (Low & Woolgar 1993) where, perhaps because of her background as a management scientist (Low et al 1996), she had difficulty penetrating the technical discussions.

This research has focused on two identifiable ways of defining a publicly shared reality that focus on software development: the technical software developer and the non-technical user or manager involved in the development. It has consciously ignored other views.

In summary, this research is based upon the recognition that for at least all practical purposes, multiple realities exist and that it is neither helpful nor useful to maintain the pursuit of a single truth. It is not helpful because by implication where one group of people believe 'X' and another 'not X' and only one of them is right, then one group must be more right than the other. It is not useful because it prevents us from examining the differences and similarities between realities and their effects on software development. This research explicitly recognises the existence of publicly shared realities and, in particular, those shared by different stakeholders in relation to the software development process.

According to Guba & Lincoln (1994) this positions the research in the constructivist paradigm.

3.3.2 Epistemology

'What is the nature of the relationship between the knower or would be knower and what can be known?' (Guba & Lincoln 1994 p108)

Within the ontological position of multiple realities, there are implications for epistemology since there is no longer the necessity to maintain the objectivity of the researcher implied by a positivist view of a single 'real' world (Guba & Lincoln 1994).

One of the problems with understanding software development is the difference between the public, rational view of development and what actually happens in real time. For example, software developers know about the value of good design but often do not undertake design activities (Parnas & Clements 1985). To investigate this difference it is necessary to get close to the development, which is not consistent with objectivity (although of course distance does not ensure objectivity (Dey 1993)). In adopting a subjective approach the roles of the researcher and researched need to be considered (Cassell & Symon 1994). In this research, the researched are involved in the construction of the knowledge. This has been achieved by making the nature of the research explicit in my conversations and in many cases discussing with the research participants emergent theories. As researcher, I have more input to the process than the researched. This is part accidental and derives from my positivist background.

3.3.3 Research methodology

'How can the inquirer (would be knower) go about finding out whatever he or she believes can be known?' (Guba & Lincoln 1994 p108)

This is again constrained by the ontological and epistemological stance adopted so that my view on reality and of research roles implies that my methodology involves the construction of realities through discussion. As Morse (1994) so clearly illustrates, there is a close relationship between the overall strategy or strategies adopted within this methodological framework and such issues as research question and outcome. This section discusses potential methodologies for the research and outlines some basic principles. The detail of the research method is expanded in chapters 4 and 5.

The constructivist paradigm implies that knowledge is constructed during the research with a view to developing an understanding rather than building a causal model (Guba &

Lincoln 1994). This is similar to Burrell & Morgan's (1979) interpretivist paradigm that is 'firmly set against the utility of a search for laws or underlying regularities in the world of social affairs' (ibid p5). Knowledge accumulation is seen to be through the 'formation of ever more informed and sophisticated constructions' rather than 'a process of accretion, with each fact (or probable fact) serving as a kind of building block' adding to knowledge (Guba & Lincoln 1994 p113-114).

With this in mind the research question is reframed to be:

'If software development is explored holistically as a human activity, what emerges?'

As indicated in section 3.2.1, the research is based on longitudinal case studies where a case study is understood to be an investigation of a 'contemporary phenomenon within its real-life context' (Yin 1994 p13). Descriptive case studies are extensively used in information systems research and are particularly suited to investigations which consider the software development process as a whole rather than focusing on some smaller element (Swanson & Beath 1988). However, in this research the case studies are used to address an exploratory type of 'what' question (Yin 1994) to develop tentative theory. Although the theoretical sensitivity of the researcher guides this theory building there are none of the theoretical propositions recommended by Yin (1994).

With the emergent interest in ethnographic work in computer science and information systems research, described earlier in this chapter, it was natural to consider an ethnographic approach to the cases. Although there are variations in the definition of ethnography (Hammersley & Atkinson 1995) a number of key elements seem to characterise ethnographic work. First and foremost is the idea of an in-depth study of life that 'involves the ethnographer participating overtly or covertly, in people's daily lives for an extended period of time, watching what happens, listening to what is said, asking questions - in fact collecting whatever data are available to throw light on the issues that are the focus of the research' (Hammersley & Atkinson 1995 p1). Within the study the researcher treats events as 'anthropologically strange' so as to make explicit what we take

for granted. The description of cultures is one of the main goals of ethnographic research (Spradley & McCurdy 1972).

An ethnographic approach was adopted for data collection but not used as a guiding methodology because the focus of this research is not on the nitty-gritty detail of the culture of software development but rather on identifying the forces in the 'bigger-picture' that shape software development.

The exploratory nature of the research question indicates that a grounded approach is appropriate for data analysis (Dey 1993). Grounded theory has already been used in information systems research (section 3.1.2) and Eisenhardt (1989) draws on it for her case study approach. However, grounded theory is presented as a scientific method (Strauss & Corbin 1990 p27) of the post-positivist tradition (Guba & Lincoln 1994) which is not necessarily consistent with a constructivist paradigm. In particular, grounded theory aims to build substantive theories by generalising through sampling different groups of people and this is not consistent with the study of a few longitudinal cases of software development.

At the level of method, however, grounded theory has much to offer this research with its emphasis on interpretation rather than just 'giving voice' to the researched. The iterative approach implied by the interplay of data analysis and data collection is very important since the complexity of the situation under study means that the focus of the study will change and develop as the research progresses (Cassell & Symon 1994, Silverman 1993).

The choice of research methodology is important because how you study something affects the phenomena that are revealed (Smith 1990, Morgan 1983). This section has considered various approaches and has determined that the research should be conducted by the qualitative study of a few in-depth longitudinal cases. Ethnographic principles should be used in data collection and analysis should be grounded in the data.

3.4 RESEARCH QUALITY

The development of new research paradigms has led to confusion over the criteria to be used to judge the quality of research (Smith 1990). Within a positivist tradition the quality of research is judged according to internal and external validity, reliability and objectivity (Guba & Lincoln 1994). Problems arise with these criteria at ontological, epistemological and methodological levels. Abandoning the notion of a single reality affects the validity criteria: how can something be judged true if there is no absolute truth? Reliability and objectivity are no longer useful measures if knowledge is socially constructed and hence subjective. The qualitative methodologies which are associated with new paradigm research are problematic for the criteria because of the subjective nature of data collection, the impact of data collection on the process under study and the difficulty of establishing external validity from a small number of cases.

Heshusius (1990) summarises a number of responses to these problems. At one end of the scale is the attitude that 'criteria are not needed' because 'if something is good, you know it' (p198). This may be a reaction against an emphasis on method and proper procedures that seems to accompany discussions of quality (Marshall 1990, Smith 1990). Nevertheless, this attitude affects the standing of the research (Silverman 1993) and would be particularly inappropriate for my audience. Once the need for criteria is accepted, one position is to parallel the positivist criteria (eg Lincoln & Guba 1985, Kirk and Miller 1986, Easterby-Smith, Thorpe & Lowe 1991, Silverman 1993). As an alternative Marshall (1990) regards research quality as a 'judgement call' and lists twenty sources of evidence. Although the reliance on positivist criteria has been questioned (eg. Guba & Lincoln 1994, Janesick 1994), I am adopting this approach because of the traditions of my audience.

In the following sections, the criteria against which this research is to be defended are discussed, distinguishing between the goodness of the research and its utility (Marshall 1990). The way in which these criteria are met is discussed in conjunction with the research method in chapter 5.

3.4.1 Goodness

Research goodness encompasses the concepts of reliability, objectivity and internal validity. Reliability is about consistency (Lincoln & Guba 1985): if I were to undertake the research again would I come out with the same results? Reliability also encompasses objectivity: if other researchers undertook the research using the same procedures would they come out with the same results (Dey 1993)? Lincoln & Guba (1985) substitute the concept of neutrality for objectivity but I prefer to borrow from Soft Systems the idea of a world-view. Within the context of the research stance explained in section 3.3, it is not reasonable to expect **any** other researcher to come to the same conclusion because of the existence of multiple realities. However, the results should be consistent with a publicly shared reality so should be shared by researchers with the same world-view. So, reliability for this research means: are the results consistent within themselves and with an acceptable world-view of which I am a part?

Internal validity is about truth: is this what really happened? Is it an accurate representation within a publicly shared reality? One way of judging this is by assessing whether the research can be defended conceptually and empirically (Dey 1993). This often translates into questions of method: has the researcher gained full access to the knowledge and meanings of informants (Easterby-Smith et al 1991)? Lincoln & Guba (1985) summarise these concerns with internal validity by using the term 'credible'.

3.4.2 Utility

In positivist terms research is useful if it is externally valid, that is the results can be generalised from the particular situations studied to a wider set of cases. Vitalari (1985 pp252/3) regarded generalisation as an impossible goal and in consequence saw it as important to provide a rich documentation of context to 'allow others to later use the research findings in constructive ways not anticipated by the original researcher'. Lincoln & Guba (1985) translate this into the concept of transferability; can the results of the research be transferred to other situations? Like Vitalari, they agree that making this transfer is the responsibility of the reader and that the writer can only provide the 'thick

description' necessary for someone to make that transfer. This view of utility is to be preferred to that of Dey (1993) who asks for representation so that more general statements can be inferred. In the context of a very limited number of case studies, it is hard to argue representation.

3.5 DISCUSSION

This chapter set out to find an approach to use in this research. The research was begun within a positivist tradition but through data collection I became aware of the inadequacies of this approach for addressing my research question. Section 3.2 explains how I came to adopt a research paradigm which is based upon constructivism in that:

- multiple shared realities are assumed, with a focus in this research on those realities which resonate with the interests of those involved in developing software;
- understanding is constructed through a dialogue between the researcher and researched;
- the research question was reframed in terms of a development of understanding as: *if software development is explored holistically as a human activity, what emerges?*
- methodology centres around longitudinal cases which are studied ethnographically and analysed using a grounded approach;
- quality depends upon consistency, credibility and transferability.

This definition of research paradigm provides a frame of reference against which the research should be judged.

Chapter 4 explains how the longitudinal cases were undertaken and summarises each case. During data collection, themes influencing all the developments began to emerge. This led to a refinement of the research question and the decision to study one case in detail. The research method developed for the analysis of that case (see chapter 5) is grounded and designed to meet the quality criteria developed in this chapter.

4 Research question

Chapter 4 explores software development as a human activity using the research paradigm described in chapter 3. Section 4.1 describes how six cases were chosen and studied longitudinally. The six cases were deliberately chosen to be different in organisation, complexity and application type but, during the studies, similarities started to emerge even though the software developments being studied were at differing maturities. Asking what was having a major effect on the development seemed to reveal a surprisingly small list of key influences. Section 4.2 summarises each case and presents the eight influences that seemed to be emerging.

The idea that it was useful to look for a relatively small number of key influences on software development was tested at a think tank of software professionals attending a major conference. Section 4.3 describes the process of the think tank and presents the nine categories that the participants of the think tank identified.

These two sections show that the concept of key influences on the software development process seems to have promise and the chapter concludes by explaining how the research question was rephrased to focus on finding a potential set of key influences.

4.1 UNDERTAKING LONGITUDINAL CASE STUDIES OF SOFTWARE DEVELOPMENT

When the longitudinal studies were started, neither the research paradigm nor method described in chapter 3 had been established. Influenced no doubt by my scientific background, I chose to study six cases in equal depth. This was in spite of being aware of literature that cautions against collecting too much qualitative data. For example, Yin (1994) recommends a single case for revelatory research; Barnes, Buckland & Brancheau (1992) found in their research that the number of cases in each field study (between 6 and

10) 'had too few participants to make meaningful statistical comparisons and too many to be conducted as true case studies yielding rich qualitative information' (ibid p330). The need for more than one case was based on the argument that the uniqueness of software developments requires literal replication (Yin 1994) and to support generalisation by the reader a variety of software developments are needed.

4.1.1 Choosing the cases

The cases were chosen using theoretical sampling (Eisenhardt 1989) to illustrate differences in:

- the type of organisation (restricted to the UK): business, size and culture;
- the type of problem being addressed by the software development: nature, size and complexity;
- the relationship between the software developers and the users of the software (eg in-house, software product, use of contractors);
- the methods, tools and technology adopted by the developers;
- the experience within the organisation of software development.

The selection of cases was also subject to the software developments being at a stage where they would progress through to some form of implementation within the time-scale of the research. Difficulties in getting access to the organisations or to the right people for extended interviews was not found to be the problem that other researchers in this area have encountered (eg Land, Le Quesne & Wijegunaratne 1992, Low & Woolgar 1993). The organisations were provided with a short summary of the research together with a commitment to anonymity for the organisations and individuals concerned. Moreover, at the outset, the organisations generally regarded the projects as good examples of forward-looking software development.

The six cases came from the five organisations summarised in figure 4.1 and varied considerably in size and ethos. I was personally involved in the first two cases, V and W, and knew organisation Z so in these three cases I had insider knowledge. I was introduced

to X by one of my supervisors and to Y by my father but their links with these organisations were slight so I approached these two cases as an outsider.

Case	Main business	Approx. size (employees)	Approx size (turnover)	Geographical distribution	Ethos
V	Management education	100	£12-14 million	One site in South East	Academic entrepreneurial
W	Specialist joinery	6	£200K	Two sites in South East	Traditional family business
X	Insurance	235	£40-150 million	Offices/agents world-wide	Competitive financial
Y	Network technologies	20	£1 million	One site in East of England	Leading edge R&D
Z	Education	3000	£200 million	Midlands with offices in Europe	Academic democratic

Figure 4.1 Organisations in the longitudinal cases at the time of the studies

The two academic institutions were included because they were actually tackling a very similar problem but on a different scale as shown in figure 4.2. There was a preponderance of information-rich systems in the selection but these were on different scales and developed using different approaches (figure 4.3).

Case	Aim of software project	Scope of development	Length of initial dev.	Est. cost at outset	Max. no. developers
V	Replace operational system	Organisation-wide (eventually)	2 years	£100Ks	4 + consultant
W	Initial computerisation	Office functions	6 months	£1K	consultant
X	New information system	Organisation-wide	2 years	£100Ks	about 15
Y maint.	Maintenance	Existing products	1 day to 6 months	variable	3-2
Y blue sky	Provide a future for the company	New product(s)	1 year	£10Ks	2-4
Z	Replace operational system	Organisation-wide, some BPR	about 7 years	£10M	about 50

Figure 4.2 Scope and size of software developments in the longitudinal cases

approach meant that I failed to interview someone with an important perspective because none of my informants thought the person was important. This may have been avoided if I had analysed my cases in depth as the data was collected but this would only have been possible if the study had been limited to one case.

For each case, data was collected with the aim of telling the story of this development, that is, what was done in the project, by whom and why. Recognising that software development rarely runs to plan, I also tracked the course of changes. A further complication was the different realities of many software developers and users, so there is not just one story to be told. Thus, in collecting the data the aim was to get the people directly involved in development to construct their view of what had happened within the project and why.

As is normal in qualitative research data was collected from a variety of sources (Van Maanen, Dabbs & Faulkner 1982) and the ‘big three’ approaches (Cassell & Symon 1994) of conversations, participant observation and document analysis were all used. I use the term conversation to reflect the informality, lack of structure and two-way nature of the in-depth interviews (Easterby-Smith, Thorpe & Lowe 1991, Mishler 1986). The emphasis was on in-depth unstructured conversations and documents in X, Y and Z with participant observation used in V and W. Ethnographic context of the developments was noted but mostly to assist me in my role as ‘an active and thoughtful listener’ (Marshall & Rossman 1995 p65).

Within the conversations, I decided to aim for rapport rather than neutrality agreeing with Mishler (1986) that these are incompatible. While Easterby-Smith, Thorpe & Lowe (1991) stress the need for the interviewer to listen without projecting opinions, I found that putting forward theories and giving examples from my own experience or the experience of the other cases was a good way to test theories and encourage people to talk. This reflects my view that the researched are active participants in the research and a sympathy with Fontana & Frey (1994) that traditional in-depth interviewing is unethical in treating respondents as objects rather than individual people. There was rarely any difficulty in

Case	Scenes (ie contacts with the organisation)			Recordings (ie conversations, documents, meetings)			
	Total	Visits	Time span of regular visits	Primary conversations etc	Secondary documents etc	Pages	Hours
V	10	9	22/6/94 – 9/2/95	17	6	65	10
W	13	12	10/1/93 – 4/10/94	14	1	39	0
X	25	13	30/9/93 - 17/2/95	32	55	648	30
Y	13	9	17/12/93 – 13/1/95	35	29	388	20
Z	58	34	20/12/93 - 17/5/95	50	86	1247	42
Total				148	177	2387	102

Figure 4.4 Data collected for each study

The next section describes the cases as they appeared during data collection including the major influences that appeared to be emerging.

4.2 EMERGING INFLUENCES IN THE LONGITUDINAL CASES

In the summaries of each case in this section care has been taken to protect the anonymity of the organisations and the people involved in the software developments. Each person has been given a pseudonym and details such as location have been changed to conceal the identity of the organisations. The glossary defines specialist terms used in the cases.

4.2.1 V – medium sized entrepreneurial academic organisation

V, a successful, well-respected business school within a UK University runs post-graduate and post-experience short courses. The courses are operated on a self-financing basis and are professionally marketed. I became involved with software development in V when, in 1991, the University I was working for won the business to train and advise them on the development of a system for the Short Course Department. I was not involved in the delivery of the initial training but subsequently acted as a consultant to the development team until Autumn 1993. I had conversations with each of the main actors about a year after the end of my consultancy contract and some follow up conversations about six months later. These conversations were all taped and conducted in a similar manner to X,

Y and Z. The following description and analysis in this thesis is largely based on these conversations but informed by my own knowledge of the development.

This development provided one of the motivations for the PhD because there were problems with the adoption of the system, illustrated by a manager deciding to call the new system ‘CUPID’ because if it went wrong it could then be called ‘STUPID’. This occurred despite best practice input from my University’s Computer Science Department including teaching on systems development.

Two departments were involved in the software development, Marketing Services run by Glenda, and Short Courses run by Lucy. Glenda had been in post for some time and had several staff working for her. Her department was responsible for marketing all courses run by the school and operated at the request of the Senior Management Team. Marketing used a database system known as ‘Speedy’, which had been written specially for V about six years earlier. This system contained a large database of potential customers (about 10,000 companies and 16,000 contacts) and was used to produce regular reports and targeted mail-shots. A fulltime database supervisor, Alice, was responsible for maintaining the quality of the data and producing the reports and labels.

Lucy had recently taken over Short Courses, which was responsible for the administration associated with the operation of short courses. Each course was assigned to a Short Course Administrator who organised the domestic arrangements for the course, took bookings and liaised with the academic actually teaching the course. Short Course administration used Word for producing lists, badges etc, and limited information about each course was input to ‘Speedy’. This included delegate details so they could be included in future mailings.

There were problems with ‘Speedy’ almost as soon as it was installed, particularly for Short Courses, but by 1990 it had also become problematic for Marketing. It was slow, running out of storage capacity and did not provide the functionality now needed by Marketing. In 1990 and 1991 much work was done on evaluating suppliers to provide a replacement for ‘Speedy’ covering both Short Course and Marketing requirements but the recommendation was not accepted by the Senior Management Team. The new Short Courses Manager, Lucy, and the chair of the IT Committee put together a case to develop an in-house system

first for Short Courses but with a view to extending it to Marketing later. This case was based on using Oracle running on Unix as this was becoming a campus standard.

The development team consisted of three people with little experience of developing large systems and no experience of Oracle and Unix. The technical manager, George, was an ex-academic who was currently responsible for 'Speedy' and had systems experience but no programming experience. The project manager, Holly, was an IT literate researcher with no experience of managing a software development. Nathan, the programmer/analyst had developed dBase systems and was aware of structured analysis techniques. To counter the team's inexperience, V sought advice from my University, which then ran courses and assisted the development. Colleagues taught the team Evolutionary Systems Development (Crinnion 1991) and Oracle. George and later Nathan also attended Unix courses.

The first phase of the development provided a system for Short Course Administration running in parallel with 'Speedy'. An evolutionary prototyping approach was adopted involving Short Courses and Marketing staff. The application was developed as a mainframe Oracle application accessed from PCs running a terminal emulator, with extracts to Word Mailmerge running on the PCs. Phase one took about nine months to develop and included a substantial delay in the delivery of the Unix machine. At the end of phase one, the chair of the IT committee left and was replaced by someone with less practical IT development experience.

Phase two replaced 'Speedy' by providing a new Marketing system integrated with the new Short Courses System. This phase included the conversion and deduplication of data from 'Speedy' and other sources. It went live about a year after phase I in the Autumn of 1993. An additional programmer, Yolande, was finally recruited during this phase in Summer 1993. Following phase II, work continued on enhancing both applications and installing new versions of Oracle and Word. In Summer 1994, the original programmer, Nathan, left the University and the project manager, Holly, moved on to other projects. George intended to retire but this was delayed pending the recruitment of an IT manager. There seemed to be a lack of interest shown by the Senior Management Team in the system's fortunes despite its apparent importance to V. There were difficulties with the recruitment

of IT staff and at my last conversations (in February 1995), no IT manager or replacement programmer had been hired and George had not retired, but a consultant from another part of the University had been brought in to help the remaining programmer.

During the development there were numerous problems with the technology. Some of these problems were due to the inexperience of the team with Oracle but other problems arose because of integration of the various technologies. There were also ongoing performance problems on both the Unix machine and the PCs. Despite the use of prototyping, not all the users were committed to the system during the development as exemplified by the choice of 'CUPID' as system name. However, by the time of the final conversations most users seemed fairly happy with the system, although it did not live up to expectations. Following the installation of phase II there seemed to be a lack of vision and impetus and less development was done than in the first two years of the project. There were difficulties integrating the new programmer, Yolande, into the team and then a loss of knowledge when the first programmer, Nathan left. On my last visit, the outlook for the system seemed more positive as the two user managers, Glenda and Lucy, were taking a more integrated and pro-active role in the systems development.

4.2.2 W – small specialist joinery company

W was the first case studied specifically for this PhD but initially it was part of an experiment to investigate the effects of method on IT strategy (Tagg 1994a). As the research method for the PhD became clearer, the experiment was not extended but this case was retained because it is different in size, scope and industry to the other cases. W is a small family business run by a husband and wife team, Robert and Heather, who I know well. They agreed to participate in the research provided I helped them with computerisation. Assistance with computerisation continues to this day but this case only covers the first twenty months. The case began with a series of working sessions applying the chosen method to W. These sessions were not taped and the notes were mainly about outcomes. As the research method changed the conversations did cover more ground but it was not possible to tape them (because of the location of the conversations and we were all

too conscious of the tape). Moreover, due to the initial contact and the continued consultancy role it was not easy to separate doing from reflecting.

W was founded by Robert in 1979 shortly after leaving furniture college. The business has been built upon Robert's expertise as a furniture maker, his skills and enthusiasm for refurbishing second hand machinery, and his ability to build up a network of customers among architects and designers. The company makes high quality bespoke furniture and specialist joinery largely for the commercial market. Jobs are very varied; at the time of the study typical examples were: individually designed boardroom furniture in fiddleback sycamore, the veneering of 60 doors in ash, and a fitted kitchen for a private customer. W employed three other craftsmen: a works manager (Donald), a trained furniture maker and a trainee. Heather, Robert's wife, was a partner in the business and managed the paperwork and finances, liaised with customers and suppliers, and designed kitchens. W also had a part-time van driver, employed casual labour as required and sub-contracted some of the kitchen fitting. The company was growing slowly.

W had no computers or fax machine but had an impressive array of specialist machinery and a large stock of fittings, veneers and specialist timbers enabling it to undertake a very wide variety of work. The small office was dusty as it not completely closed off from the machine floor. Paperwork was kept to a minimum at W; many quotes were verbal or handwritten, invoices were typed by Heather who also kept the handwritten ledgers, and the payroll was done by an old lady on the farm (where part of the workshop was based).

I became involved with W because Robert was considering buying a portable computer. He thought that a computer would be useful for word processing and invoicing and wanted a computer to help with cutting lists by performing some of the calculations and working out the best cutting layouts. He had some ideas about keeping records of jobs on the computer but reservations about the amount of time that this would take. Robert was considering a portable because it was neater, used less space than a full-size computer and would be convenient to take work home, but was worried whether it would survive the dust.

In approaching a problem of this nature, it can be difficult to scale down appropriate methodologies to the needs of a small business. The starting point and language is wrong,

and they can take too long to complete. Process Quality Management (PQM) (Ward 1990) is an intensive, structured approach which uses a critical success factor approach to identifying IT opportunities. PQM was successfully used at W and this led to the development of an approach to help SME's make business critical computing decisions (B•CDD – see Bennett et al 1996).

PQM was used with W over three evening sessions with Robert and Heather to identify IT priorities. To counter their lack of IT experience, a demonstration was arranged to show them how the software identified could be used to assist with the processes. The result of the demonstration was a recommendation that W should consider buying an entry level Apple Macintosh and using it initially for word-processing. The case for the purchase of a computer was not clear on the basis of the PQM analysis, and although W were interested they decided they could not afford it. This decision was partially influenced by their works manager, Donald, having a motorcycle accident and breaking a leg two days before the demonstration. Following his accident, as Donald was unable to stand for several weeks, he suggested that he could usefully spend the time getting the invoicing sorted out on his own computer. He proposed bringing the computer to the office and lending it to W on a long term loan. The computer, a 386 PC with a dot matrix printer, had Windows, Word for Windows, a shareware accounts package and spreadsheet. Donald had experience of the accounts package but little experience with the rest of the computer.

For a period, the accounts package was used to issue invoices but the software was unsuitable for W's business and they stopped using it and produced invoices using Word, with a list of outstanding invoices maintained in Word for keeping track of debtors. The computer gradually became an invaluable tool for Heather. They acquired a laser printer as part-payment for a spraying job and set up templates for use with new pre-printed stationery. As I stopped my observation she had just bought Sage accounting and payroll packages partly because their accountants were recommending it. Throughout this time Heather became more familiar with using the computer but Robert did not use it.

4.2.3 X – large world-wide insurance company

X was the first case that was conducted as a longitudinal study especially for this research. My original contact was with the IT department in the parent company. They suggested that I should look at the Multinational system they were developing for X because it could be typical of the way future systems would be developed. X-Group is the parent company of a large worldwide insurance company that was formed in the nineteenth century. X was formed in 1991 to bring together units in the group specialising in complex multinational insurance packages for specific clients. At the time of my first contact, the first phase of the system was about to be installed. I collected documentary evidence about the origins of the system and observed the development of further phases for the next seventeen months.

Multinational Operations was a department of underwriters in X responsible for underwriting programmes of insurance for large multinational companies. These programmes would typically involve policies being issued in a number of countries to cover a variety of risks. In the UK, the Multinational Services Department was responsible for the administration of programmes and ensuring local offices issued policies. Local offices of other companies in the X-Group or partner companies handled claims and premiums but the money was ceded back to the controlling office. The Multinational Claims team was responsible for handling large claims. Much of the business was reinsured to spread the risk of losses occurring at any one location. The nature of the business was long term with accounts being renewed annually and relatively few opportunities to quote for new business. This was in complete contrast with other parts of X-Group writing domestic or motor insurance.

Teams of underwriters wrote multinational insurance programmes in a number of countries but at the time the system was started the main office was in the City of London (about 400 programmes being managed by 25 underwriters) with smaller offices in USA (about 80 programmes) and Holland (about 40 programmes). The Multinational Services and Claims groups were based in Essex with other servicing teams from X-Group. I had conversations with the developers, underwriters and managers of Multinational Operations

and Services in the UK. I did not speak to anyone in Claims because it was not until the end of my observation that their potential importance as users of the system became clear.

Before the formation of X, each group writing multinational programmes had used local IT systems. These were not designed for this type of business and much of the work remained manual or used locally developed PC-based systems. In the UK, a PC-based system had been developed by an underwriter, Keith, using a database package, Q&A. This was in use in Multinational Operations, Services and Claims in the UK and Holland.

The Multinational system was developed by an IT development team (GIS) in X-Group head office in the North of England. The system originated from an IT Strategy study followed by a Requirements study and was developed using an in-house evolutionary prototyping approach. The system was a client-server application using Sybase and Powerbuilder. Initially it was installed on local area networks in UK (Essex and London), USA (New York, Chicago, Los Angeles) and Holland with transfer of information between each country's databases occurring overnight over ISDN. The whole system was initially installed on PC servers and Windows clients running a Novell network but later Unix machines replaced the PC servers. In most cases the PCs and local area networks were installed as part of the project. Email and word processing were also installed and for many of the users this was their first exposure to Windows.

Phase I of the system was jointly managed by David, manager of Multinational Operations and Gordan, project manager in GIS. David and Keith were seconded full-time to the project. The system was developed in under a year using fortnightly prototyping workshops in London involving representatives from Multinational Operations, Services and Claims. Phase I concentrated on the data input and transfer aspects of the system with only a basic set of reports. When phase I was installed it was much bigger than originally envisaged but was generally well regarded although it ran too slowly. During development the team realised that the database structure was badly designed although it evolved from an original entity model. Following phase I there was a lull in development while the developers recovered from the push to install phase I and senior management in X decided their priorities. At this point David left the project and was replaced by a more

junior person, Jenny, working part-time. The technical team leader of the project also moved to another part of X-Group.

Phase II ran in parallel with analysis work on a system for another department in X, Marine, and a Financial Accounting sub-system for Multinational. During development of Phase II, the live system suffered from database crashes caused by a problem with Sybase (which Sybase eventually solved) and some other errors. The first part of phase II was essentially a redesign of the database and client software. It went live six months later and suffered from many small bugs and some further intermittent Sybase problems. These problems caused a loss of confidence in the system and in the prototyping methodology. Eventually the PC servers were replaced by Unix machines which seemed to solve the Sybase problems, the management of the development team was strengthened and a more traditional structured approach to development was introduced.

Throughout the period of my observation at X, the system suffered from data quality problems. In the UK and Holland, basic data was converted from the existing underwriter Q&A system. In the UK it was checked and updated on a rolling basis by underwriters as the programme came up for its annual renewal. In the USA data was entered from scratch. When I stopped observing, 20 months after phase I had gone live, the data in the UK had not all been updated and the system was about to be relaunched in the USA because it was not being effectively used. The main reasons put forward for this were the culture amongst the underwriters and the problems with the system. Multinational Services found the system had less functionality than the Q&A system it replaced. Developments to upgrade the functionality for Services were planned when I stopped observation.

4.2.4 Y – small network technologies company

I made contact with Y through my father and his involvement in computers in education but neither he nor I knew the organisation. Y was founded in 1978 by Keith and another partner to produce add-on boards for Acorn computers. Through Acorn's involvement in the education field, Y became one of the first companies in educational computer

networking. In 1983, the other partner left and Keith was joined by his brother, Neville. Neville and Keith were joint managing directors with Neville looking after the financial and commercial aspects and Keith the technical aspects of the business. Keith and Neville both had civil engineering degrees but while Keith was the deep thinker with the ability to see commercial opportunities in emerging technologies, Neville was more committed to the more mundane growth of the company.

When I visited the company it was still a family business but had grown to about 20 staff. It was based in a small industrial park in South-East England. Their main product was an extensive range of networking components for schools using Acorn machines. Their aim was to provide simple complete networking solutions designed to meet the particular requirements of schools, and the company both designed and produced their own hardware and software. The previous year had been very stressful with changes in the schools market brought about by changes in funding arrangements and the growing influence of PCs. As a consequence of this just before my first visit, they had a strategic planning session with the entire company. Out of this had come a new organisational structure and the decision to set up a 'blue sky' project involving Keith and two of the four engineers from R&D. The objective of the 'blue sky' project was to produce an idea for a product to develop the future of Y beyond the life of the current products.

The company suggested that I should follow the progress of the 'blue sky' project and the maintenance work on the existing products for my research. I was invited to attend research meetings approximately every six weeks and had conversations with most people in the company. I also spoke to Keith and Neville on each visit.

Observation of the 'blue sky' project was more about innovation than software development. Early on they decided on a focus on ATM, which was emerging as an important high-speed network protocol to support multimedia applications requiring the transport of large quantities of data. For the first five months they followed a number of side-tracks learning about technologies and building links with other interested parties. At this point they decided to build a basic video-conferencing demo for an exhibition in October and a more sophisticated demo for the main education exhibition, BETT, which

occurs in early January. During this period all members of what became known as the ATM team also spent a lot of time on other maintenance activities.

Between the July and September meetings, the direction of the ATM team changed as Y became a partner in an Interactive Television Trial based in South-East England. Y was designing and building the ATM switches; development work began on 22 August and the first phase of the trial went live on 30 September. After the first phase, Y were involved in a bigger trial and in December, the brothers sold most of the educational aspects of Y to allow them to concentrate on the new ATM developments. This deal had been under negotiation for about four months. In this way the core of Y reverted to a small R&D biased company and the staff (both those staying with Y and those leaving) seemed pleased with the way the year had turned out.

The maintenance team consisted of Ron, who had newly been promoted to manage the team, a young rather volatile engineer, Oscar, a number of part-time software engineers and the ATM team as and when required. When the observation started, Y's main products were not as reliable as they should have been and the maintenance team was very overloaded. In February, Oscar was supposed to be starting on a major new print server with a view to having a beta release for testing during the summer holidays and the product available in September. This project continued to slip and by early December there was still no sign of it, Oscar had decided he was ill, some of the company thought he was lacking in motivation and Sales had no confidence in ever getting the product. In conversations with Oscar and Ron I helped them to see that a critical problem was that the project was much larger than anything Oscar had tackled before and he could not explain the internal structure of the print server. With the blockage identified, development went well and by BETT, in early January, a cut down product was available. Unfortunately by this time they had largely lost the market for it and Neville accepted that the difficulty in managing the print server development had been commercially expensive.

Although I did not observe much software being built at Y, it illustrated some of the problems encountered by those building a software product. It was interesting that similar sorts of influences emerged in what was quite a different case.

4.2.5 Z – large academic organisation

Z is a large UK University based in the Midlands with a number of regional centres. In 1991 Z's IT department published a proposal to redevelop the student records system as the mainframe needed replacing. The proposal was in line with their technical 'open systems' strategy to deliver administrative systems via a network of over 1000 Windows PCs acting as intelligent workstations. Originally the student record system was used for all student admissions but over the years a number of alternative systems had been developed to handle different sorts of programmes; by 1991 there were 17 different ways of admitting students. Changes in funding arrangements in Higher Education meant that there were no longer funding reasons for keeping all these separate approaches and with an increasingly competitive market in Higher Education there was a need for Z to be more flexible and customer-centred.

The technical proposal was followed by a six-month user strategy study published in January 1992 which proposed the development of new systems to support all Z's customers. Ernst & Young evaluated these proposals and recommended that Z develop an information systems strategy to ensure that the proposed systems would meet the future strategic requirements of the University. A number of strategy studies were identified and the one for customer support began in September 1992 and concluded in July 1993 having involved about 150 users in workshops or interviews. The strategy defined project C starting in September 1993 and finishing five years later at a total cost of about £10 million. Project C was divided into eight business areas (based on a CRUD entity analysis) and three overarching projects each involving many new applications. A project charter for each area and plans for the whole project were part of the strategy. It was recognised at the outset that this was a huge project, which was 'far more complex and bigger than we dare reveal' (Paul, project C Director). It was estimated that over half the University would be directly affected by the project. Paul was part of the general administrative services of the University and had been involved in the information strategy work. Reluctantly, because he recognised the risks, he agreed to lead project C.

I began observing project C in December 1993 when development had just started. I concentrated on the largest business area covering marketing, student enrolment and sales of materials, which was scheduled to deliver some applications in 1996. In addition I collected some material about a much smaller, separately funded project for research degrees which was being developed under the auspices of project C in advance of the main areas. This project was regarded as piloting many of the techniques and technologies of project C but was not really representative because it was so much smaller and had been initiated outside the information systems strategy. I observed project C directly until May 1995 with 50 conversations and the collection of 72 documents and continued to collect published materials until the first phase had been implemented in 1996. The conversations included all the senior people directly involved in the development, users from two faculties (Business and Maths) and many of the users and analysts involved in development, but because of the stage of the project, I only had one conversation with a designer and none with programmers.

Users of information systems in Z had become increasingly unhappy with the way in which the IT department developed systems and project C was developed using a new approach involving users much more closely. The PRINCE project management approach and Ernst & Young's Navigator were used for the information systems strategy. As part of the strategy a small study (2 users and 2 IT staff) selected a methodology for project C. They decided to employ a small consultancy firm to use rapid development techniques to help them develop an in-house development method based on an Information Engineering approach, and to use PRINCE for project management. The method was developed during a 5-day workshop but unfortunately the consultancy did not live up to expectations and the method was not fully defined. PRINCE was found to be over cumbersome and not aligned with the democratic decision making process in Z, where many decisions have to be ratified by Senate comprising all academics and other stakeholders. Both the method and the integration of PRINCE continued to be developed throughout the first year of project C. A basic CASE tool was purchased for the project and used in conjunction with an existing data dictionary package. The project was developed by users and IT staff working very closely together. In many activities a small sub-team of a user and an IT person would

work together on the development of a model or a screen prototype. A variety of workshops were used throughout the project as a way of consulting with users more widely.

The project got off to a slow start in November 1993 as a result of delays in the University taking decisions and difficulties in recruiting full-time users quickly. By January 1994 the marketing, enrolment and sales area was already one month behind schedule and the fee accounts area was two months behind. At the outset, project C had been seen as an agent for change within the University but no business process re-engineering activities had been incorporated into the development method or schedule. This affected the schedule for business analysis of enrolment and fee accounts but the results were innovative and did result in substantial changes in the way the University operated. The slowness of the University decision making process affected both areas, particularly enrolment. The difficulty in resolving issues raised during business analysis affected options selection because the method required that one phase should be finished before the next began.

Options selection for marketing, enrolment and sales began in August 1994 and was complete by September 1994. It was not clear that any real options were being presented and so it took about a month for the options report to be signed off so design could begin. This resulted from problems with the method and as a consequence the method was revised and this stage became outline solutions design. Despite the amount of time, care and resource that had been devoted to the method selection and development, it continued to be a problem for the team working on enrolment. This was because the method was new to the institution and they were the first group using it seriously.

Throughout 1995 the enrolment applications were developed and a new graphical user interface for project C was established. In parallel work was done on the overarching projects: a products database that all project C applications could use; a new technical infrastructure; and a standard management information application for both PCs and Macs to query all project C data. The implementation of phase one of project C was recognised as large and a senior person was appointed to manage it in December 1994, although implementation was not scheduled to start until January 1996. Implementation involved changing publications, redesigning forms, reorganising offices and moving 140 staff, plus

95 days business and system training. With the first phase of project C all student details would be captured in the new system but existing applications using that data would need to be used for at least a year. Routines were developed to clean and migrate data onto the new system (including resolving 19000 students with multiple identifiers) and copy new data back to the old system.

Unfortunately the system was not ready for implementation in January 1996 and to avoid a year's delay, an interim system was put in place. This involved the temporary use of an existing system modified to capture student details using the new processes. The implementation was rescheduled for April/May but was further delayed until late May with the main new system coming on line on 4 June 1996. Overall the reported reaction of administrative staff seems to have been neutral but as a consequence of the delays the entire project had to be rescheduled and the mainframe computer had to be retained for a further year.

Throughout the project the team worked hard at communicating with the many users who were involved. Apart from workshops and interviews during analysis there were presentations and road-shows to the regions that summarised the project and sought feedback. A newsletter was produced approximately every three months and widely circulated. Despite the effort put into communication there continued to be a lot of ignorance and misconceptions about the project. It was particularly difficult to involve academics in project C and none were seconded to the project.

Project C set out to develop new systems, which would support business change, using a professional development approach involving users fully in the process. In this they largely succeeded; the project was properly managed and the project team were responsive to criticism. Despite this the question was raised more than once: was it necessary to have such a large project? Would a decentralised approach have been quicker, cheaper and more effective?

4.2.6 Emergent key influences

During data collection common themes began emerge even though the projects were different and at various stages of maturity. Towards the end of data collection, in February 1995, the following tentative themes were identified.

a The power of the individual

Individuals seemed to be very influential in each of the cases. In W, the idea of computerisation came from Robert's desire to have a portable computer because it was neater. In X, Gordan was very important in holding together the team; the users seemed to believe in him even with all the problems of phase II. In Y, Keith was the brains behind the products but also seemed to provide direction to the engineers. On one of my visits, one of the senior engineers spilt some coffee on the floor and asked Keith what he should wipe it up with.

b The impact of chance

Many of the developments seemed to be influenced by lucky (or unlucky) coincidences. The most striking was the way in which W decided to go for a PC with a particular collection of software because their works manager was knocked off his motorbike. At the time, it felt as though the way in which Y ended up developing the ATM switches was the result of a lucky set of coincidences. In Z, the project manager of the enrolment area recalled how a chance remark late on in business analysis led them to remodel the registration process. It felt as though my visits were also chance events; the people I was talking to would take the opportunity to discuss something they had been mulling over and as a consequence take perhaps different action.

c The essential ingredient - change

In Z the need to change business processes was an integral part of the project although process change was not included in the method or in the early plans. Projects were also affected by changes in the organisation. The research degrees area was not part of the information systems strategy in Z but was included in project C because changes in legislation meant that there were funds available to develop a system.

d The importance of design

Design was not often explicitly discussed by many of the participants despite some direct questioning, but still seemed important. The problem that Oscar had with the print server in Y was in part because he could not explain its internal design. The need to rewrite the first version of X's Multinational system was because the database design had been neglected during the prototyping of phase I.

e The influence of the organisation

The organisation seemed to have a big influence on each of the developments. The decision-making process in Z made it very difficult to use the PRINCE project management method. It also delayed the resolution of issues and probably explained why the IT department had taken to making the unilateral decisions it was criticised for. In X the culture among underwriters was widely regarded as the reason why they were slow to load their data into the new system.

f The difficulty of communication

Communication is regarded as important in software development but the cases illustrate how difficult it is to be successful. Z produced regular, readable newsletters but many people had claimed not to see them despite wide circulation. Moreover some people derided the newsletter because they felt the style was too light-hearted. Both Y and X illustrate the difficulties of communicating with technical people. In X, Gordan was unaware that the entire system had effectively been rewritten in phase II and in Y, Neville did not have a technical discussion with Oscar about his problems with the print server.

g The need for a vision and belief

One of the problems in V was maintaining a belief amongst all users in the system despite the use of prototyping. Conversely one of the notable things about X was the way in which the users would describe all the problems they were having with the system but end by saying that it was a good system. An overall vision for the system was also important in the cases. In phase I of X, David had a very clear vision for the system and this helped them to achieve much in a short time, but in V the vision and drive seemed to evaporate after phase II and progress was slow.

h The problems of technology

Both X and V suffered from performance problems with the system running too slowly. In V, inexperience with both Unix and Oracle meant that they were ill-equipped to address the problems.

The idea that a relatively small number of fundamental things could have a serious effect on any software development was tested by a think tank of software professionals in March 1995 as described in the next section.

4.3 THINK TANK AT OBJECT TECHNOLOGY '95

The annual Object Technology conference is noted for the innovative and practical nature of its sessions. For this reason it attracts an international audience of software practitioners and technical managers who come predominantly from industry (252 out of 279 participants in 1995). Think tanks were introduced for the first time in 1995. Each of the five think tanks lasted 75 minutes and was tasked with producing a poster for display during the remainder of the conference. I convened a think tank on 'Critical success factors in object-oriented development' at the conference and this section summarises the process and the outcomes.

The aim of the think tank was to identify and classify the key factors that impact on the success of an object-oriented software development with the objective of providing guidance to team leaders of object-oriented developments along the lines of: 'The most important things to get right are ...'. The think tank was limited to object-oriented developments because of the subject of the conference, but the aim was to explore the idea emerging from the case studies that there are a small number of things that have a major impact on software development.

As illustrated in figure 4.5, the think tank was designed to distil the experience of the participants, to identify and categorise the critical factors for object-oriented development, representing the categories on a poster as one line statements encapsulating the experiences of the think tank participants.

A draft of section 4.3 was actually written by the convenor using the flipcharts produced during the session; circulated to most of the participants, and amended following their comments. Twenty-nine people participated in the think tank. As shown in appendix B, the participants mostly came from a wide range of commercial organisations.

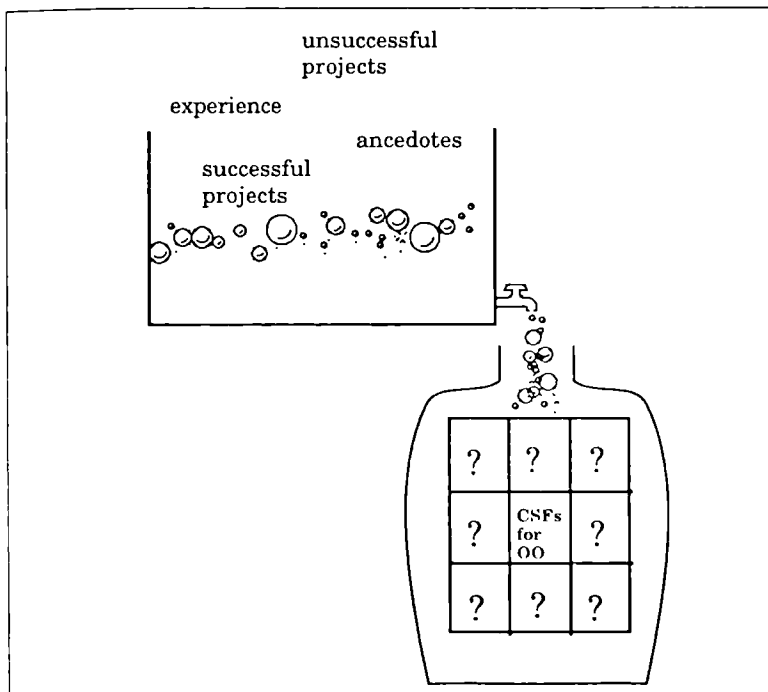


Figure 4.5 Think tank approach

4.3.1 Organisation of the think tank

The success of the think tank depended upon drawing on participants' experience of developing software. Participants were alerted to this in the conference materials and were asked to review present and past projects (preferably object-oriented) and consider the questions:

"The thing(s) that really saved this project were ..."

"The thing(s) that really messed us up were ..."

before attending the think tank. The first few minutes of the think tank were spent with participants individually listing the factors that were critical to the success or failure of their projects. They were asked to think of underlying causes rather than the result. For

example, if ‘late changes in requirements’ were identified as a particular problem, the participants were asked to consider why there were late changes and why the changes caused problems. They were also asked to rank their list in terms of importance for future projects.

The second stage of the think tank, which lasted about thirty minutes, was to categorise the participants’ factors. The convenor presented a list of categories emerging from the case studies (see figure 4.6) as a useful starting point for the discussion.

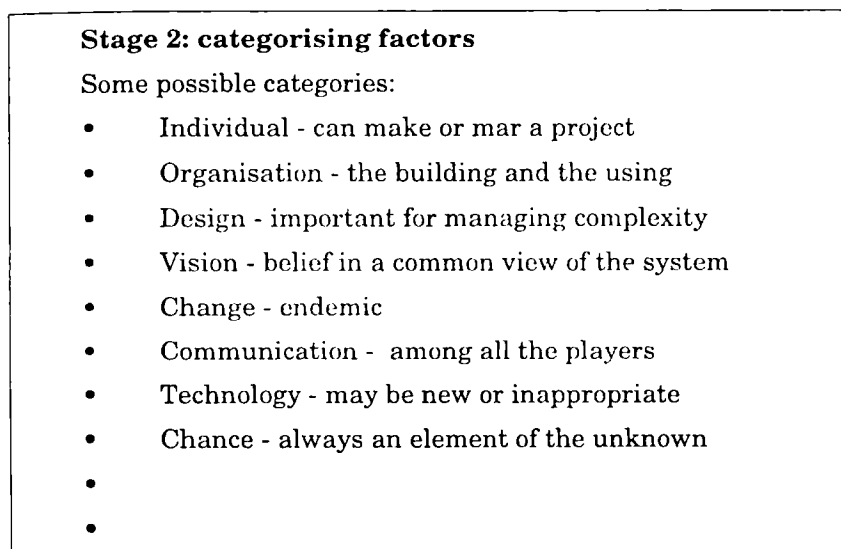


Figure 4.6 Categorising factors: the starting point provided in the think tank

In the subsequent debate these categories were used but vision was divided into two categories covering ‘investment/commitment’ (with something of a user bias) and ‘technical vision’ and organisation was renamed ‘culture’. Additional categories were added covering: team (size, dynamics, composition), management & control, process & planning, education, magnitude & scope. Time constraints meant that the categorisation was rather ‘rough and ready’ so that there was clearly some overlap between categories. The discussion ended when each participant felt they were able to categorise each of the factors they had listed. For each category, participants were asked to indicate if they had factors for that category; the counts, as they were listed during the think tank, are shown in figure 4.7.

Category	Votes	Category	Votes
Individual	9	Education	8
Team	16	Communication	12
Investment/Commitment	14	Culture	11
Design	12	Magnitude & scope	6
Technical vision	11	Change	7
Management & control	15	Technology	15
Process & planning	12	Chance	2

Figure 4.7 Think tank categories with number of participants who had factors for the category

The final stage of the think tank was to form sub-groups to discuss the nine most popular categories (that is those scoring 11 or more). This number was chosen partly based on the number of participants but also because this seemed to reflect a natural break and some of the smaller categories were subsumed in the larger categories.

A table was designated for each of the chosen categories and participants were asked to work on the category they felt was most important (with a proviso that no sub-group should exceed four). Interestingly, the last table to be filled was the sub-group discussing 'Design'. Each sub-group was asked to work on flip-charts recording the factors contributing to a category and the ways the factors impact on object-oriented development. It was suggested that they characterise the experiences of the sub-group using a graphical representation (such as a mind map, bubble diagram, influence chart, cartoon or rich picture) if possible. Each sub-group was asked to produce a short one line summary message for inclusion in the think tank poster. This part of the think tank was very animated and lasted for about 30 minutes until the end of the session. The think tank poster consisted of the nine summary messages emerging from each of the sub-groups.

The next section describes the categories which emerged during the sub-group discussions and concludes with the relevant summary message.

4.3.2 The emergent critical success factor categories

These sections are based closely on the flipcharts produced the sub-groups during discussions on factors, amended by comments received subsequently from those that contributed. The categories are discussed in the order of the votes in figure 4.7.

a Team (16 votes)

There is a need to dovetail the individual role or focus with a common team vision so that both personal roles and team objectives are clearly understood. In team forming, individual hidden agendas need to be exposed so that the context can be understood and the decision making process identified. Mutual trust and respect is very important because it encourages openness in communication. The team will be rewarded, not the individual.

The composition of a team must be balanced so that there is a spread of skills and an avoidance of dependence on 'key' members. The size of a team needs to be carefully considered. The role of the team leader is one of leadership, co-ordination and arbitration.

In forming and managing a team, the approach should be to **determine common aims and objectives then actively seek cohesion.**

b Management and control (15 votes)

The discipline of cost control and monitoring is important for success. This requires good estimation (which is difficult), planning and replanning and the setting of budgets. It is also important to be able to learn from previous experience and take corrective action. One of the problems with object-oriented development is the lack of more than two or three years of experience. The small units achieved through the use of objects is helpful but there is a dearth of knowledge about estimation for object-oriented development.

In general, management issues are no different with object-oriented development so it is necessary to beware of the hype and manage expectations carefully. Motivation is just as important as is the need to evaluate the real worth of consultants and sub-contractors.

To summarise the important message when managing an object-oriented development is:
forget the hype - most management issues are the same, object-oriented is jam the day after tomorrow.

c Technology (15 votes)

Good, appropriate technology can lead to success for a project by providing rapid results, infinite potential or a springboard for future development. If the technology is 'good' it frees the team to concentrate on other issues.

In contrast the negative influences of technology can be very stressful by raising false expectations or giving rise to technology overkill. The choice of tools is important in this respect. Inappropriate technology can arise from unresolved strategic issues, a lack of evaluation, enforced standards, or cost constraints and usually results in increased requirements for education.

The solution is a **timely implementation of an appropriate technological foundation.**

d Investment/Commitment (14 votes)

Without investment and commitment to the development there will be inadequate resources provided for the project. The resulting system will tend to focus on political rather than business advantage. In consequence there will be a lack of clear requirements and objectives and people's productivity will tend to be low.

Without commitment, communication will be poor particularly if things are going wrong. In particular there will be a lack of feedback during development to check that the system is actually meeting user needs. There is also a risk of the withdrawal of IT and user management support and developing systems is hard enough without that!!

To be successful, **a two-way, whole-hearted commitment between business and IT (stakeholders and developers) is vital throughout the project.**

e Process and planning (12 votes)

There are two elements in this category: the development framework and risk management. The development framework needs to be simple, well-understood, repeatable and support throwaway prototypes. Risk management requires a separation of the management process from the design process. Risk is minimised by using an evolutionary delivery mechanism with short firm deadlines and by encouraging user feedback.

In order to plan and manage risk it is necessary to estimate both time and money. Metrics need to be collected but we need to use them with care. Experience helps but how do we get it?

The overall advice is that the **process must support phased iterative contract review covering commitment, time/money estimates and scope.**

f Communication (12 votes)

Bad communication prevents knowledge transfer while good communication lets knowledge filter through to the appropriate people. Management layers are essentially about communicating information between various people and co-ordinating this information. Good communication allows proactive management; if things go wrong, communications start to break down.

Successful projects are those where team members know project goals. Each project involves thousands of decisions made by people at low levels. Without clear aims, the consequences of these decisions cannot be predicted (although they often cannot be predicted with good communications).

The aim of methodologies is to enable sharing of knowledge/information (via requirements specifications etc.) between users, team and individual team members in a clear and concise way. Methodologies should also allow for the storage and retrieval of all those little bits of information that are generated during the life of a project.

Without good communications each team member and user are working on different projects.

g Design (12 votes)

One of the key factors identified in the design category was experience which enables the developer to know where to focus and avoids obsessive concentration on minor features. It is also important to have the right mindset - to think in the object-oriented way. One of the problems with the 'object-oriented paradigm continuity' is that it can seduce the developer into going directly to coding from analysis; the result is a disaster. It is important to have a good physical architecture; the "object soup" is unmanageable without a design process. Good communication with other technical areas is also a key factor. This was summed up by the need for the following mindset:

Openness to new approaches.

Openness to different points of view.

Openness to others contributing to design.

Being conscious of the Object-Oriented way of thinking.

h Culture (11 votes)

Software development needs a forgiving culture in which code can be thrown away. It is important to encourage responsibility for designs or code so that they are produced to a high quality on schedule. However, this must be within an 'ego-free' development environment in which ownership is depersonalised so that people talk about 'the' design not 'my' design. Formal mentoring is helpful and will lead to informal networks of advice.

The culture in a big team is quite different to a small team but the overall aim should be to foster a **learning culture with shared knowledge and tribal memory**.

i Technical vision (11 votes)

Technical vision is a statement of 'how we do things around here'; that is the 'rules' about how the logical is mapped to the technology. For a technical vision to work, it must reconcile project and organisational goals and it must be shared. In order for there to be a shared technical vision, there needs to be exactly one messiah. The messiah must be part of the organisation, omnipresent, a communicator, enthusiastic and a recognised authority.

In summary: **visions need messiahs.**

4.3.3 Conclusions

The participants at the think tank showed interest and commitment both during the session and afterwards. Considerable agreement was reached on the factors in the relatively short time of the think tank and despite lively debate. The wording of the factors in the previous sections are generally those used by the participants on the flip charts. Most participants regarded the think tank as an interesting and useful experience and nearly all participants identified themselves and asked to receive the paper. There was a reasonable amount of interest shown in the think tank poster, although this was the last think tank of the conference, so the poster was only displayed for two hours.

The ease with which the participants arrived at the categorisation of factors and the almost complete agreement that the final paper received, provides good support for the concept of key influences on software development. It is interesting to note how most of the categories are not technical and are not specific to object-oriented development. This reflects the opening plenary lecture of the conference given by Adele Goldberg entitled 'When to blame the objects' which was largely about project management issues.

4.4 DISCUSSION

This chapter has illustrated the difficulties of undertaking qualitative research. Although there are difficulties in collecting the data, in practice this is where the challenge starts. There is so much of interest in the data that making sense of it and drawing out some useful results seems at first sight to be impossible. This chapter has explained how the idea, that there are a small number of key influences on software development, emerged from a holistic analysis of six longitudinal cases. The idea was tested at a think tank of practising software developers. (The word influences is adopted following Seddon (1995) to emphasize the role of people and down play causality.)

One of the interesting aspects of the analysis of the six cases was that the same key influences emerged whatever the maturity of the software development process: whether the software was being conceived, constructed or delivered. These aspects of software

development often involve very different players and require different skills so it would perhaps be surprising if they were influenced by the same things.

To focus the analysis of the longitudinal studies a research question which incorporates the findings from this chapter is:

If software development is explored holistically as a human activity, what are the key influences on the conception, construction and delivery of software?

This question retains the holistic approach identified in chapter 3, but within that context, it asks what are the key influences on the different aspects of software development. In order to investigate this question it is necessary to undertake a detailed analysis of the case study material. This analysis needs to construct the realities of the case and analyse influences on the conception, construction and delivery of the software from both social and technical dimensions.

Undertaking such an analysis in a way that would satisfy the requirements for research quality outlined in chapter 3 is a substantial task. For this reason only one of the longitudinal cases studied was subjected to detailed analysis. X was chosen for detailed analysis because of the six cases, it was the most appropriate. The data was collected about all three aspects of software development and it was a large complex development using modern approaches and tools. Chapter 5 explains how the X case was analysed and the results of the analysis are presented in chapters 6 and 7.

Chapter 8 discusses the results of the detailed analysis of X and compares them with the findings of this chapter. A summary analysis of the other five cases described in this chapter is undertaken in chapter 8 to investigate the wider applicability of the findings.

5 Research method

Chapter 4 established the need to analyse the longitudinal case X to identify key influences on the conception, construction and delivery of the Multinational system. This chapter explains how that analysis was undertaken.

One of the challenges of the qualitative approach to research is that there are few established detailed methods and none that are entirely appropriate for this research (chapter 3). Consequently as this chapter describes and discusses a method was developed for the analysis. An explanation of the method and a rationale for the main elements is contained in section 5.1.

In parallel with the data collection, a piece of software was developed to assist with the analysis. Section 5.2 describes how the research method was crystallised and enriched through the development of the software. This section compares this software with established qualitative data analysis software.

One of the outstanding issues with qualitative analysis software is how such software can be used to present the data and methods of analysis. The approach adopted in this thesis is to use hyperlinked text presented via a web browser. The data and complete details of analysis are contained on the accompanying CDROM. The construction and content of the CDROM is covered in section 5.3.

The issue of research quality discussed in chapter 3 was of prime concern when developing a method for analysing X to identify key influences on software development. Section 5.4 discusses how the method supported, by the software and the presentation of results on CDROM, has addressed these issues.

5.1 METHOD OF ANALYSIS

One of the problems for qualitative analysis is that the procedures are less well established and by necessity more fluid than for quantitative work (Miles & Huberman 1994). Dey (1993) likens it to doing a jigsaw where the researcher cuts up the pieces and, I suggest, decides what kind of picture they are going to build. However, there is a conflict between being so ad hoc that the results are anecdotal and not grounded in the data, and being so mechanical that 'the serendipitous - the puzzle that if attended to and pursued, would provide a recasting of the entire research endeavor' (Marshall & Rossman 1995 p111) is ignored.

So for qualitative research, it is normal for the research method to develop as the research progresses, but it needs to be sufficiently visible to satisfy the requirements of research quality. In particular, the reader needs to find the results credible (see section 3.4.1). For this thesis readers are likely to come from the disciplines of management, information systems and computer science. Although the use of qualitative research is growing in management and information systems, it is little used in computer science and for these readers to find the research credible, the research method needs to be structured and clearly explained.

The method used for analysing the X case study is summarised in figure 5.1. Section 5.1.1 explains the origins of the method and the choices that were made in developing it and sections 5.1.2, 5.1.3 and 5.1.4 describe how the method was applied to the analysis of X.

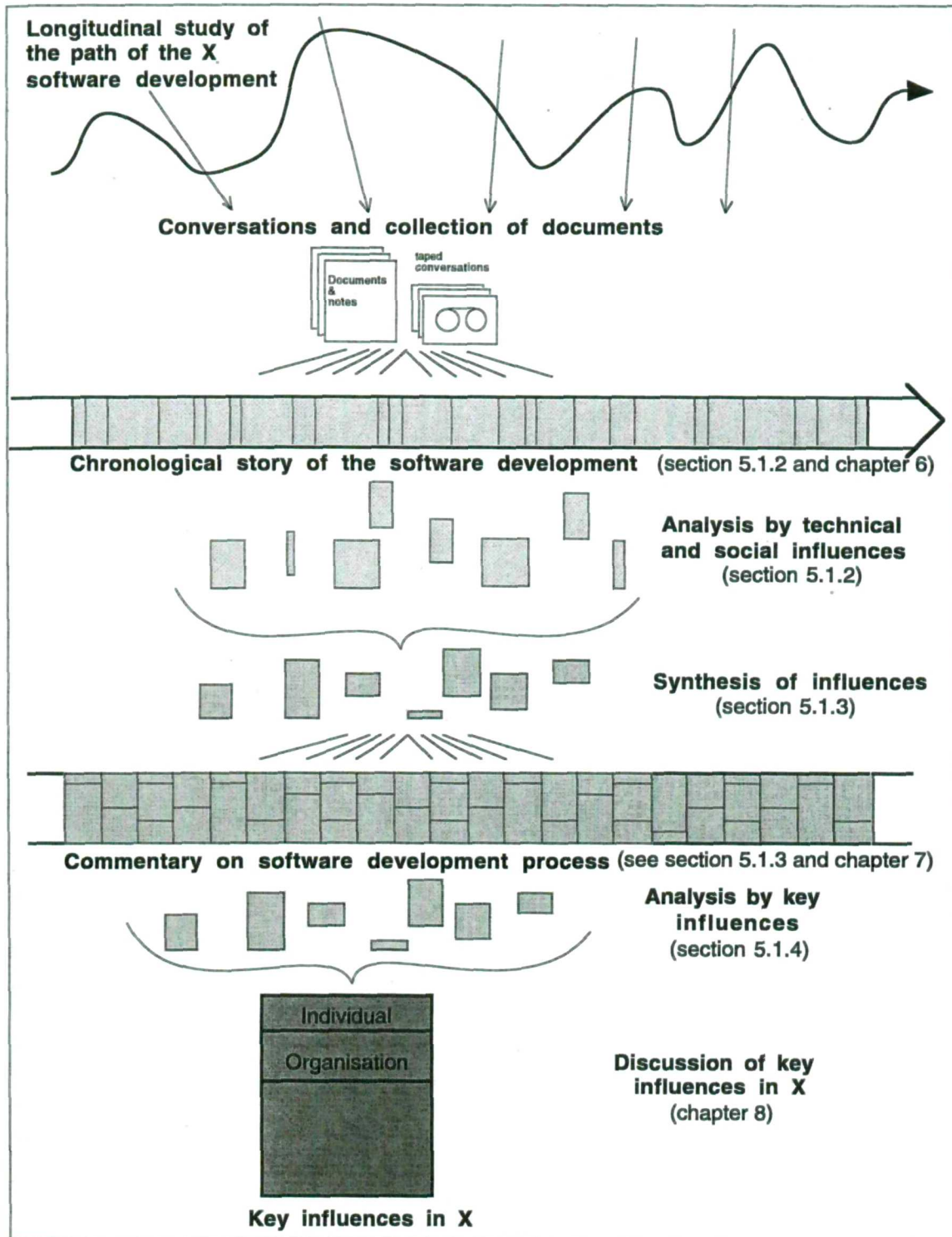


Figure 5.1 Research method used to identify key influences on software development in X

5.1.1 Development of method for case study analysis

Human beings are notoriously bad at making sense of large pieces of narrative, as there is a tendency to be selective about the episodes that are remembered (Miles and Huberman 1994). So there is a need for a systematic, comprehensive but flexible approach to making sense of qualitative data (Tesch 1990). However few published studies describe the analysis in detail with a 'huge chasm' separating data from conclusions (Eisenhardt 1989). The identification of key influences on the software development in X is a mixture between case study analysis and grounded theory. Case study analysis applies because X is a case and there is a need to understand it as a whole because the essence of key influences is that they have an effect on the whole process of software development not just on isolated incidents. As already explained in chapter 3, grounded theory concepts are appropriate because of their rigour and usefulness in deriving ideas from the data where there is little pre-existing theory. Neither approach provided a complete method but both were used in developing a new method.

The literature on single case analysis (eg Yin 1994, Stake 1995, Creswell 1998) does not provide much detail on how to arrive at a description of the case in a methodical manner. In order to identify key influences it was necessary first to establish the variety of views of what happened in the software development. This involved combining the views of reality presented by the different stakeholders at different points in the development. Studying the transcripts of conversations and case documents made this difficult because each only presented one view of reality. To address this problem without imposing the researcher's view of reality, a chronological account was developed based largely on the stakeholders' own words (see section 5.1.2). This chronological account was used for subsequent analysis rather than the original source documents, to ensure that these views of realities were equally considered.

In contrast, grounded theory provides detailed procedures and techniques (Strauss & Corbin 1990). However, applying a grounded theory approach to analysing the chronology would have been problematic because identifying the key influences required a holistic consideration of the case rather than a study of the minutiae. This is not a problem with

grounded theory per se but meant that it was difficult to provide the necessary audit trail. Moreover, due to its post-positivist epistemology, grounded theory tends to be used where a theory is to be built about cause and effect. In this research, the aim is to identify the key influences affecting software development, not to develop a theory of the impact of these influences. A multi-tier approach of categorisation, memoing and synthesis was developed from the ideas of grounded theory.

The chronology was categorised according to the immediate or direct influences exhibited in each extract. Segmenting the data in this way (without losing sight of the whole) and using categorisation to reorganise the pieces in order to 'contrast, compare, analyze and bestow patterns' (Miles & Huberman 1994 p7) is common in qualitative analysis. An interpretational rather than structural analysis (Tesch 1990) was adopted to focus on the content of the data rather than its form. The language of software development is rich and worthy of study as illustrated by Low & Woolgar (1993) but this was not the purpose of the research. Although Silverman (1993) argues that a focus on the structure of interview talk is not preferring form to content but rather 'gaining access to a cultural universe and its content of moral assumptions' (p108), I felt that a focus on language would alienate some readers and was not necessary for me to interpret the chronology. In developing an organising system (Tesch 1990) for the categorisation, the concept of dimensions was introduced to highlight the different ways of looking at the data.

Viewing the chronology through the focus of each influence provided sufficient insights to allow a commentary to be written explaining the events that had significant impacts on the software development. This commentary was developed using synthesis memos written about each direct influence as described in section 5.1.3. This technique, of producing an account that draws on the use of memos in grounded theory (Strauss & Corbin 1990), is part of the process of analysis (Dey 1993) and an integral part of qualitative research (Meloy 1994).

The resulting commentary is a much more synthesised treatment of X than the original chronology and because it presents a more holistic view of the case it may be categorised according to key influences. The key influences were then further explored by considering

all the events in the chronology relating to a particular key influence as described in section 5.1.4.

5.1.2 Constructing the chronological story

In order to produce a believable account the audience needs to be able to have a thorough understanding of the case (Lincoln & Guba 1985). This account needs 'to portray the world in terms of the constructions that the respondents use, seeing the world "through their eyes" as it were, and expressing their constructions in their own natural language.'

(Lincoln & Guba 1985 p365) Moreover the account should not be interpretive or evaluative so that readers can come to their own interpretations. As the way in which people tell stories affects the reality that is portrayed (Riessman 1990) the story of the software development was constructed almost entirely out of direct quotations from conversations and documents rather than the more normal qualitative account with short embedded quotes (Creswell 1998). A chronological ordering was adopted as recommended by Miles & Huberman (1994) because this was judged to be interpretively-neutral.

a Selecting and ordering the text units

The first stage of the analysis was to read every item collected during the study, divide it into evidence extracts and categorise each extract chronologically. The size of the extracts varied from one sentence through to several paragraphs and was determined by the need to retain context while referring to a single point in the chronology. On the first pass, all text that was at all relevant to the story was included. This included almost the whole of the transcripts of taped conversations, notes from untaped conversations and extracts from most case documents. The source of each extract was recorded so that it could be expanded or clarified during refinement.

The development of the chronology categories and the categorisation of the data were iterative parallel activities which resulted in 1110 items of evidence categorised into 8 sections as shown in figure 5.2.

Chronological category	Evidence extracts	No. words	Trimmed		
			Evidence extracts	No. words	% words
Organisational context	87	13,076	73	8335	64
Strategy study (Nov 91 - May 92)	47	4,251	35	3001	71
Multinational Requirements study (May 92 - Aug 92)	49	4,812	37	3485	72
Phase I (Sep 92 - Jun 93)	155	27,565	141	16,502	60
Use of phase I (Jul 93 - Mar 94)	99	24,941	92	14,099	57
Phase II (Sep 93 - Mar 94)	162	29,949	144	17,644	59
Phase II goes live (Apr 94 - Aug 94)	191	43,389	181	24,158	56
After phase II (Aug 94 - Feb 95)	320	82,495	268	39,750	48
Overall	1,110	230,478	971	126,974	55

Figure 5.2 Major chronological categories and the amount of evidence collected for each category

The evidence extracts were iteratively refined until they provided a minimum chronological story. This involved ordering the items within a chronological category, omitting duplicates, and omitting parts of items. Words were trimmed from items to remove redundancy, repetition (except where it was important to reflect the meaning), to make the conversational text easier to read and rarely for reasons of confidentiality. Occasionally words were added to make the meaning clearer. Words were not actually removed, just hidden, so that it was easy to change the trimming as the development of the chronology proceeded. Although most of the questions from the conversations were trimmed, care was taken to only do this where the question did not appear to lead the answer. (As an aside, when reviewing the transcripts, it was interesting to note how often the respondent appeared to ignore the question and pursued what they thought was of interest. Many of the tapes contained long monologues punctuated only by sounds of encouragement from me. This reflects how respondents were enthusiastic to tell me their view of what had happened, what might happen and why.)

'IThe review is done, as you can see, and, in addition to that, I**there's been a review of the underwriting data quality on the system.** I{Yes, that was under way - no, just about to start.}I **That has been completed now.**I

{Oh, right. They did that really quickly then, didn't they?}

Yes, they did it in about four months really. It was very quick. What they did was they took a sample of the programmes and policies, went out to Holland and they haven't done America, so when I say it's finished,I **it's been done in Holland and here.**

I{They decided that America wasn't far enough on to do anything?} Yes,I **and basically America for X is a big problem, not just on the systems side, but all aspects of the business. It needs sorting out. The people have changed and there's new management over there, besides which, the timing wasn't right to go and look at America.'**

Figure 5.3 Evidence extract 00719 from an interview with Gordan illustrating the trimming of text (text in bold was not trimmed, text in {} were my remarks)

Figure 5.3 illustrates the trimming in practice. This interview occurred after the data audit was complete and there had been many conversations covering the conduct of the audit so Gordan's comments on how the audit was done added nothing. This extract was interesting because of Gordan's view of the problems in America. The earlier parts of the extract were retained to keep the context of Gordan's remarks.

The refinement continued throughout the subsequent analysis and reduced the number of words by almost one half and the number of extracts to 971 items (see figure 5.2). These remaining items consist almost entirely of direct quotations, and together tell the chronological story of the Multinational Development in X. The chronological story is included in its entirety on the CDROM with each evidence extract on a separate page. Trimmed words are indicated by '...' and added words by {}.

The construction of the chronological story is the culmination of the filtering that occurred during data collection through:

- the choice of data to collect by timing of visits, choice of people to interview, questions asked, documents collected;
- the loss that occurs during transcription through non-verbal communication, problems with recording, errors in transcription (Poland 1995).

In undertaking the trimming described above, I was very conscious of this filtering and trimmed progressively as I sought to represent the realities of the story. The focus was to capture what happened and why from the world-view of a specialist, user or manager involved in software development in a way that was readable. My experience as software specialist, user and manager helped to appreciate these world-views.

All subsequent analysis was based on the chronological story, which continued to be refined during analysis. The summary of the story presented in chapter 6 was only written to help the reader of this thesis access the material.

b Categorising the chronological story

The next stage of the enquiry was to analyse the chronology by asking the question of each extract: does this illustrate some influence (or absence of influence where it might be expected) on the software development process? At this stage, influences were considered that were close to the data (known as direct influences) so avoiding the necessity of looking at the wider picture for key influences. Three aspects of the software development process, conception, construction and delivery (see section 1.1) were considered. As the categorisation progressed, two dimensions emerged: technical influences concerned with the actual activities of software building and social influences concerned with the people and their environments. Considering each extract from both dimensions helped to maintain a balance between considering the development from a technical and a non-technical viewpoint. Thus the research question was finally refined to:

If software development is explored holistically as a human activity from both social and technical dimensions, what are the key influences on the conception, construction and delivery of software?

The categories within each dimension were developed from the data but also informed by theory. The main categories in each dimension are shown in figure 5.4.

Each evidence extract was categorised by at most one technical and one social influence. This limitation was imposed to encourage a focus on major influences and assist the development of independent categories. If necessary evidence extracts were divided and occasionally duplicated if they illustrated more influences. The categorisation of each

evidence extract is included on the CDROM. The CDROM also includes a breakdown of all the sub-categories.

Social influences	
Individual	184 evidence extracts
Team	86 evidence extracts
Organisation	280 evidence extracts
Environment	25 evidence extracts
Technical influences	
Process	230 evidence extracts
Tasks	271 evidence extracts
IT environment	136 evidence extracts
System characteristics	171 evidence extracts

Figure 5.4 Major direct influence categories

5.1.3 Constructing the commentary

The direct influences were used to focus attention on the things that shaped the progress of the software development at X. From these a commentary was developed explaining why the development progressed as it did.

The evidence extracts for each direct influence were considered in chronological order and memos written answering the question: *what were the effects of this influence on the software development?* The evidence extracts for each influence were considered in order to produce a synthesis of the effect of each influence on the software development process. A total of 486 memos were written during this synthesis explaining why the software development in X progressed as it did.

These memos were categorised according to the event that affected the software development process in X. From this categorisation a commentary on the software development process was written which explains the events which affected the conception, construction and delivery of the software. The commentary is organised around a series of

questions (see figure 5.5). This organising structure of events and questions derived largely from a consideration of the data although it was influenced by an understanding of software development.

Software conception or inception?	
Why build a system for Multinational?	27 memos
What affected the choice of system boundary?	20 memos
What affected the choice of systems architecture?	28 memos
What affected the choice of process model?	24 memos
Software construction or evolution?	
What affected the functionality that was delivered?	52 memos
What affected the internal design?	39 memos
What affected the reliability of the system?	55 memos
What affected the schedules?	46 memos
What affected the composition of the development group?	50 memos
Software delivery or use?	
What affected the use of the system?	89 memos
Why was the quality of the data poor?	56 memos

Figure 5.5 Major categories in the commentary on the software development in X

The commentary is contained in chapter 7 and on the CDROM where it is linked to the synthesis memos.

5.1.4 Identifying key influences

The key influences arising from the X enquiry were arrived at by categorising the commentary contained in chapter 7. Each event was considered and a key or underlying influence identified. Where an event appeared to have more than one key influence, the categorisation was reconsidered and either the categories were reshaped or the event was rewritten as two separate events with different underlying influences. The categorisation started with the key influences identified in chapter 4 and proceeded through an iterative process that involved:

- identifying, refining and describing key influences;
- reviewing events categorised to the same key influence;

- refining the description of the events (by re-reading memos and evidence).

This process was continued until there were no changes to the key influences. The influences which emerged are shown in figure 5.6.

Key influence	Total events	Critical events	Important events	Potentially important events
Individual	15	11	3	1
Organisation	17	9	7	1
Method	20	5	12	3
IT profession	8	3	4	1
Belief	4	3	1	0
Resources	7	1	5	1
Domain	7	2	5	0
Technology	6	1	4	1
Design	2	1	1	0
Change	3	0	2	1
Vision	1	0	1	0
Communication	2	0	0	2
Total	92	36	45	11

Figure 5.6 Number of events categorised to each key influence

Each event was further considered to determine whether the influence was critical, important or merely potentially important in affecting the software development in X. The explanation of these terms and the categorised commentary is included in chapter 7.

Finally each key influence was considered in turn to determine how it affected the software development in X. This was undertaken by considering each event categorised by the key influence, as described in chapter 8. The possible effects on the software development in X, if stakeholders had had a better understanding of the key influences, were also considered by reviewing the summary of chapter 7 and thinking about each influence. This discussion is also included in chapter 8.

5.2 SOFTWARE SUPPORT

Software is increasingly used by researchers to support qualitative analysis by providing facilities for data management, coding and retrieving, theorising and theory testing, display and audit (Weitzman & Miles 1995, Miles & Huberman 1994, Richards & Richards 1994, Dey 1993). The use of software can enhance existing methods of qualitative analysis and provide new opportunities (Dey 1993). However the developers of one of the most used pieces of software, NUD*IST, realise that software can place 'unacceptable restrictions on analysis' (Richards & Richards 1994 p445) and the debate continues today (eg Morison & Moir 1998). These restrictions can be implicit in the software or can arise from the state of mind the software engenders in the researcher, for example an avoidance of data that is not in machine readable form, or forcing a sequential ordering onto an essentially iterative process. As with all software, the key is to view it as a tool not a panacea (Dey 1993). A piece of software was developed in Hypercard specifically to support this research. Developing the software myself prevented it from limiting the research process and also enriched the method by helping me to clarify my ideas (Tagg 1996b).

5.2.1 Development of Hypercard software to support method

Initially Hypercard¹ stacks were used as a data management tool to keep track of sources and to record facts about visits to the company (on scene cards), data collected (on recording cards) and the people involved (on pseudonym cards in the pseudonym stack). As the research method developed, the stack was extended to include functionality to support the construction, analysis and synthesis described in section 5.1 (by adding evidence and category cards and making a similar synthesis stack). Each evidence card contains text (an extract from a transcript, notes or a system document) which supports the development of

¹ Hypercard uses the metaphor of a stack of cards very much like index cards. Sets of cards within a stack can share a background and thus contain the same fields and buttons. Hyperlinks can be made

the research argument. Each piece of evidence may be categorised on several dimensions with for each dimension a hierarchy of categories. In the synthesis stack the memo cards contain memos written about the evidence that refer to specific evidence cards using hot text links (Stanley 1992). Each memo card may also be categorised using a hierarchy of categories. Figure 5.7 shows the relationships between the cards in the Hypercard stack (see appendix C.2 for examples of cards).

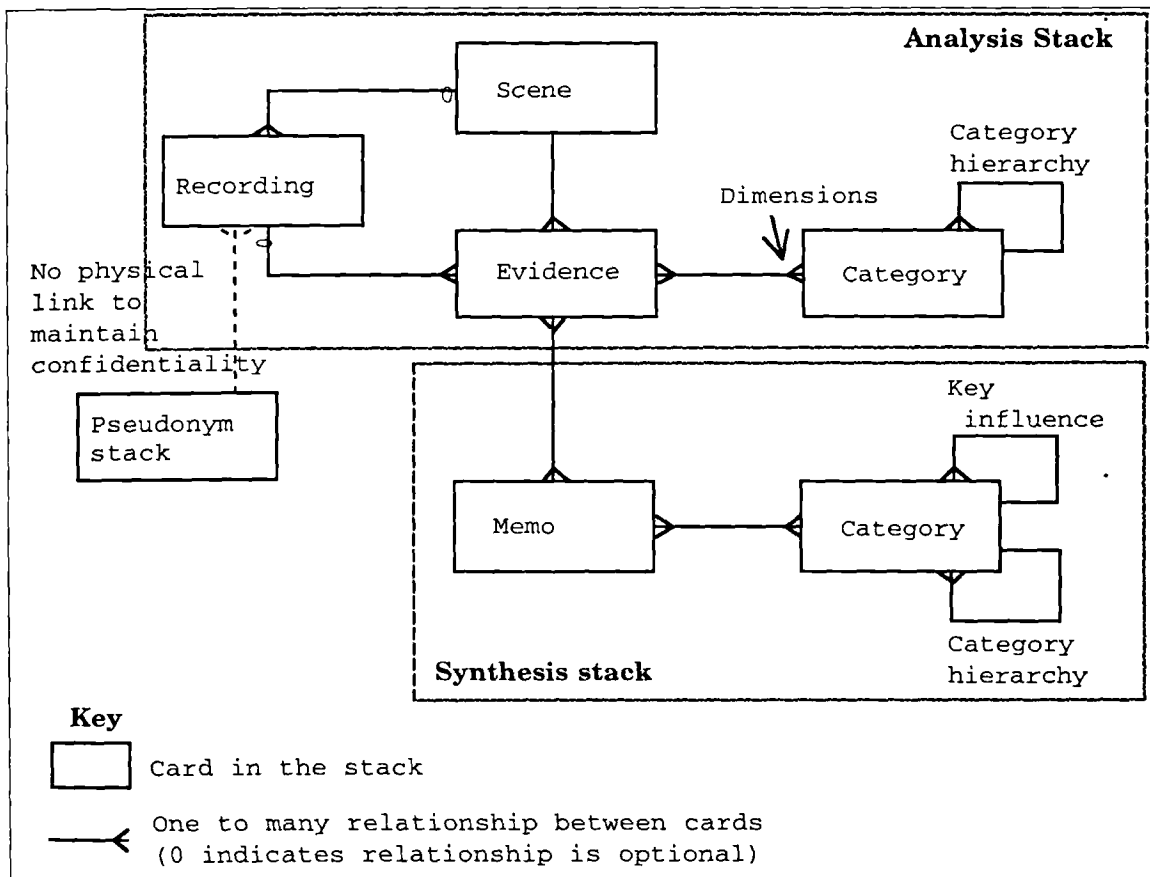


Figure 5.7 Relationship between cards in the Hypercard stacks

Functionality was built into the stacks to maintain and investigate the information on the cards and the relationships between the cards. There was a particular emphasis on the flexibility of categorisation as required by the grounded approach. Re-categorising evidence or memos is very easy as is the structuring and restructuring of the category

from a button or field on a card to a card in any stack. Linking and lots more besides can be controlled by an event-driven programming language.

hierarchies (making it possible to review everything coded to a category and recode it as necessary). Information may be extracted from the stack in a variety of orders and at different levels of detail to support the reading and rereading of the data when presented from alternative perspectives. Rich Text Format codes may be added during export to maintain the structure of the information in the export file via Word styles.

During the construction of the chronological story, each recording in the X case was considered in detail and items of text or notes added on evidence cards. These were categorised on a chronological dimension as explained in section 5.1.2. The trimming of the data was achieved by introducing a facility on the evidence cards to display the text in full or in trimmed format. The trimming was identified by inserting “|” into the text at the start and end of the portion to be trimmed. Whenever the evidence was exported to Word (or to the CDROM) the trimmed portion was replaced by “...”.

The summary of the chronology in chapter 6 was produced by summarising each category and exporting to Word using the chronological categorisation as headings and sub-headings. The facilities in the stack to code each evidence extract were used to develop the categorisation of the chronology on social and technical dimensions. Of particular importance in the development of this categorisation was the ability to recode easily all evidence coded to a particular category.

The analysis of technical and social influences (described in section 5.1.3) was completed by sorting the evidence on each of these dimensions and exporting to Word. The synthesis memos were written by reading this document. The synthesis memos with references to the evidence extracts were entered into the synthesis stack and categorised according to their contribution to events affecting the software development process in X. The commentary was written in the stack and exported to form chapter 7.

The analysis of the commentary for key influences described in section 5.1.4 was completed in the synthesis stack although this required some additional facilities to be added because it meant categorising the commentary categories. One of the advantages of developing one's own software for analysis is that this kind of extension is relatively easy to add. The

table produced in section 7.4 was generated from the synthesis stack and the discussion on key influences in section 8.1 was developed in the stack.

Both the analysis and synthesis stacks were extensively used to develop the CDROM described in 5.3. A number of additional sections were added to the synthesis stack specifically for this purpose.

5.2.2 Comparison of stacks with other qualitative software

There are a number of other Mac-based coding-oriented qualitative data analysis packages but for detailed comparison purposes two well-respected examples are considered:

NUD*IST, one of the most frequently used packages (Weitzman and Miles 1995) written by Lyn and Tom Richards, and Hypersoft developed in Hypercard by Ian Dey (Dey 1993).

This comparison uses the versions of Hypersoft and NUD*IST (version 3) that were available when the stacks were originally developed.

The software is superficially more similar to Hypersoft because it is based on the same underlying model (Hypercard) and includes specific fields for 'face-sheet variables'.

However, the categorisation hierarchy is much closer to the indexing facilities provided by NUD*IST although NUD*IST does not directly support the concept of dimensions. The flexibility provided for categorising and re-categorising is much more akin to Hypersoft although the flexibility of NUD*IST has since been enhanced. Search and retrieval options are included but largely use the basic Hypercard facilities (as does Hypersoft) and are not as extensive as NUD*IST. This reflects the relative unimportance of theory development in the research method.

Unlike both Hypersoft and NUD*IST there is no direct representation of the original text of transcripts or documents in the stack although evidence may be ordered according to source and the untrimmed version viewed. Thus during analysis I was not limited by the need to enter whole documents or transcripts at the outset. One consequence of this is that it is possible to order the evidence extracts in many different ways. Although NUD*IST

allows data to be retrieved according to the coding the text is always ordered by document so it would not be possible to use NUD*IST to construct the chronology.

The control of printed output provided through the use of preference cards and in-line control characters is much more sophisticated than either package. Much of the material in chapters 6, 7 and 8 and the CDROM was produced directly from the stack.

During the period of analysis there have been extensive changes in the software available to support qualitative analysis but NUD*IST remains one of the best known packages.

Although it has developed considerably during the period of this research neither NUD*IST (version 4 or 5) or NVivo² provide the facilities to undertake the analysis as described in this chapter. They allow the recoding of data coded to a node that is really essential for satisfactory category development but the ordering of fragments of text is not possible in either package without losing context. The only piece of software that is known to provide the facility to attach a numerical value to individual text fragments is winMAX³ through the use of weights. It is not clear that this would actually allow the text to be retrieved in a specific order. The output facilities are limited in NUD*IST and quite awkward to use in NVivo despite the use of rich text. There are no facilities for export as web pages in either package although these are available in atlas.ti⁴ but with limited use of hyperlinks. NVivo does provides multimedia and some hyperlinking capabilities with a free player, but these facilities would not easily allow the development of the links described in the next section.

² NUD*IST Vivo is a qualitative data analysis package developed by QSR International released in May 1999. Version 4 of NUD*IST was released in 1997 and version 5 in September 2000. (www.qsr.com.au)

³ winMAX is a qualitative data analysis package developed by Prof Dr Udo Kuckartz (www.winmax.de)

⁴ atlas.ti is a qualitative data analysis package developed by Thomas Muhr of Scientific Software Development (www.atlasti.de)

5.3 DEVELOPMENT OF CDROM

The CDROM that accompanies this thesis is an integral part of the submission because it allows the reader to access the chronological story as told by the stakeholders and confirm the credibility of the findings. Section 5.3.1 explains why it was decided to submit part of the thesis on CDROM and how the CDROM was created. Section 5.3.2 describes the content of the CDROM, its connection with this written thesis, and how it may be used.

5.3.1 Development of an electronic version of case study results

During the writing of the results of the case study research I was troubled by the writing style adopted in much qualitative research. It seemed that much of the writing was in the style of the authoritative narrator supplemented by carefully chosen quotes to meet the requirements of a 'thick' description by adding richness to the write up (eg Sauer 1993a). This was troubling because it implied there was a single reality as discovered by the author. I also thought that it was very easy to use quotes to enrich the writing but that this would do nothing to demonstrate the veracity of the writing. In the same way that it is very easy to lie with statistics so it is easy to find a quote to prove a point particularly when you have thousands of words at your disposal. To address these concerns, I wanted to tell the story of the Multinational Development through the words of the participants, but the result (about 127,000 words) was too long for a traditional thesis.

In my own reading of qualitative research I was frustrated by the level of detail in the text where the author's selection of content did not match my own interests. The definition of research quality adopted for this thesis (see section 3.4) requires that the reader should both believe in the research and be able to make their own generalisation. To do this the reader needs detail but too much detail will 'be incompatible with the logistics of communication' (Stake 1995 p124). Who is the reader? Stake distinguishes between the model and the empirical reader and recommends writing predominantly for the model reader. This thesis is aimed at the examiners but if the results are to be useful, the case

study report also needs to be accessible to users, managers and specialists in software development; typically the people interviewed in the course of the research. This implies a multitude of model readers with different interests and levels of understanding about software development.

A further problem with the traditional research report is the linear form. A large software development is complex and forcing a description of it into a linear presentation implies an order and consequently some priority of legitimacy. For example, a user and developer will frequently describe the same event differently. Whichever view is presented first will convey to the reader a message about the ultimate correctness of either view.

Coffey, Holbrook & Atkinson (1996) suggest that the representation of ethnographic writing has become contested, and there has been experimentation with the literary form used for presentation. Despite this, much qualitative work is still presented in a linear form even when published electronically (Slack 1998). Coffey, Holbrook & Atkinson suggest that the time has come for the exploitation of hypertext and hypermedia in ethnographic work. They rightly highlight the non-linear nature of hyperlinked documents and suggest that readers can 'become authors of their own reading'. Thus the use of hypertext can address the need to provide varying levels of detail for different model readers and avoids some of the disadvantages of a linear form. For this thesis there was no reason to use multimedia because the collection of data concentrated on text and although the tapes of conversations were still available the nature of the research did not warrant making the tapes accessible to the reader. The next two sub-sections explain how a hypertext version of the case study was developed.

a Use of web technology to represent research

Although Coffey, Holbrook & Atkinson (1996) suggest the use of hypermedia in preference to more traditional forms of qualitative data analysis software, they do not suggest how this should be achieved. A limited example using web hyperlinks is contained in the online paper by Opie (1997). Although the hypertext package, Hypercard, was used for the analysis it was not considered accessible enough for a presentation of the case study. Hypercard has a player that is freely available but is only usable on Mac platforms.

For this thesis a web-based approach to hypermedia was adopted partly because of its portability. A set of web pages written in HTML⁵ can be saved on a CDROM that can be read by a browser running on a Mac or PC. With a bit of care over the use of more advanced features of HTML, it is possible to write pages that can be viewed by most versions of Netscape Navigator or Internet Explorer running on either platform. Another advantage of the markup language HTML is that it is straightforward to use and relatively easy to generate.

b Building the CDROM

This section describes how the CDROM was generated from the Hypercard stacks. It cites examples from section 5.3.2, which provides an illustrated description of the structure of the CDROM. The simplicity of HTML makes it easy to generate but it does not provide direct support for some desired features: layout, styles and links. Layout on HTML pages is provided through tables and frames but early browsers support neither of these features. Some pages would be very difficult without tables (particularly the time line) so tables were used but frames were not, meaning that the pages can be read by version 3 and above of Navigator and Explorer. Styles help to provide consistency across the site but are not supported by version 3 of the browsers. This capability was achieved by using subroutines to write aspects of the HTML.

HTML only provides a simple link to either the top of another page or to an anchor inserted in the middle of a page. The link can be from a piece of text, an image or part of an image using a client-side map. This simple link was used to represent three types of links within the case study web-site: menu, navigational and hyper links. A frame is frequently used to provide direct access to certain key pages from anywhere in the site. These menu links were achieved using a set of links at the top and bottom of each page (see for example figure 5.13). The HTML for these links was generated using a parameterised subroutine allowing the 'Help' link to go to a different page depending upon the context.

⁵ HTML – HyperText Markup Language is used to specify the layout of web pages.

Navigational links are used to allow the pages to be navigated in different ways. For example, the evidence extracts, which are pieces of text from the transcripts of conversations or case documents, can be read in four different ways: in the order in which the data was collected, chronologically according to the event discussed, by the social influence exhibited in the text, or by the technical influence exhibited in the text. The navigational links are indicated using images of arrows on the web pages (see figure 5.8).






	start of navigation of evidence pages
	start of navigation of synthesis memo pages
	next page in navigation
	previous page in navigation
	end of navigation – up to start of navigation

Figure 5.8 Navigation symbols used in web pages

The navigation links were generated by adding additional fields to the hypercard stack and developing routines to sort the cards in the correct order and store the links. On the CDROM, hyperlinks are normally represented on the web pages as text links although there are maps with hyperlinks for the organisational structure, research method (figure 5.21) and structure of the site (figure 5.10). Hyperlinks were added wherever there were references in the original cards that might usefully be followed up. All links were translated into HTML by keeping the name of the web pages and anchors within pages similar to the numbers used in Hypercard to identify cards.

Routines were used in both Hypercard stacks to generate the links and build the pages. Each evidence extract and synthesis memo became a separate page to facilitate navigation (see figure 5.14 for an example of an evidence page and 5.18 for an example of a memo page). As the category cards had a single method of navigation, they were converted into structured pages with an index at the top and an anchor for each card (see figures 5.11 and 5.12 for examples from the story page that contains the complete story category hierarchy). This makes the pages easier to read but allows hyperlinks into the middle of the page. These routines were initiated from a special page in each stack as illustrated in figure 5.9. Approximately 1600 pages were built using this approach; only two pages in the web-site ('Index' and 'Research method') were written by hand.

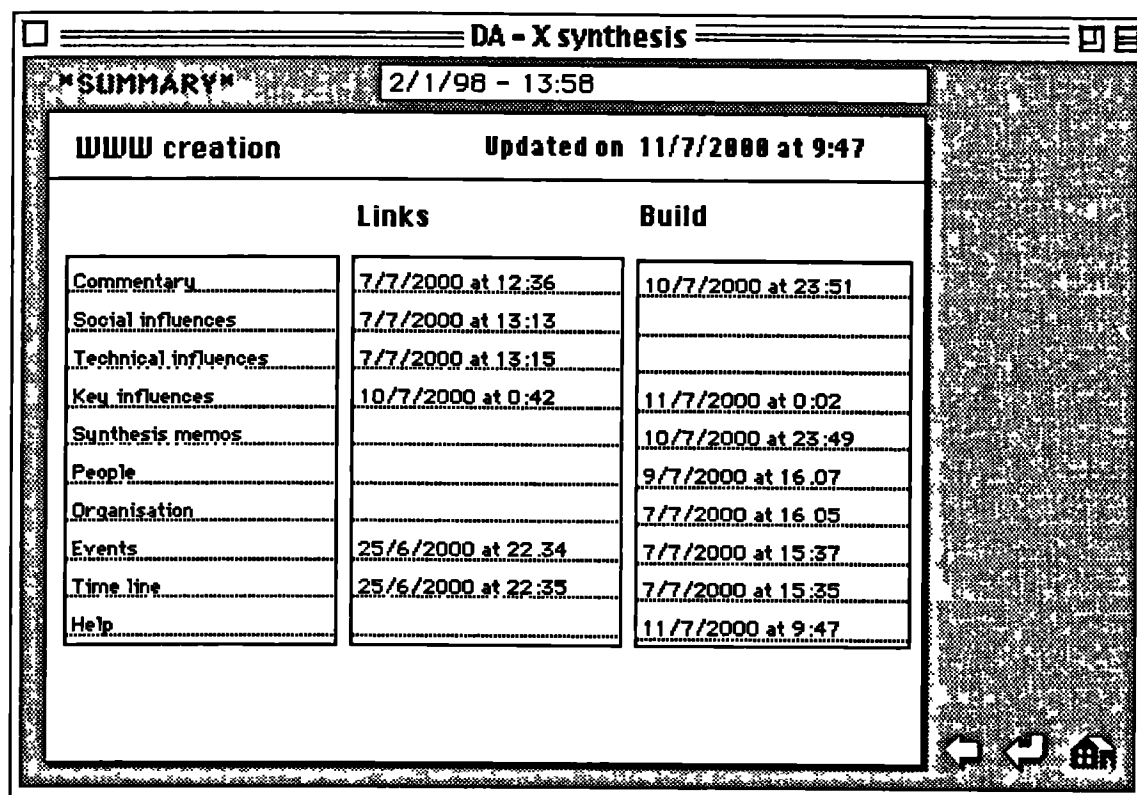


Figure 5.9 Card controlling web page build in Synthesis Hypercard stack

The next section explains the structure of the web-site on the CDROM and how it can be used.

5.3.2 Structure of CDROM

The CDROM contains the web-site and a short 'readme' file (in Mac and PC versions) which explains how to access the web-site. The structure of the web-site is as shown in figure 5.10.

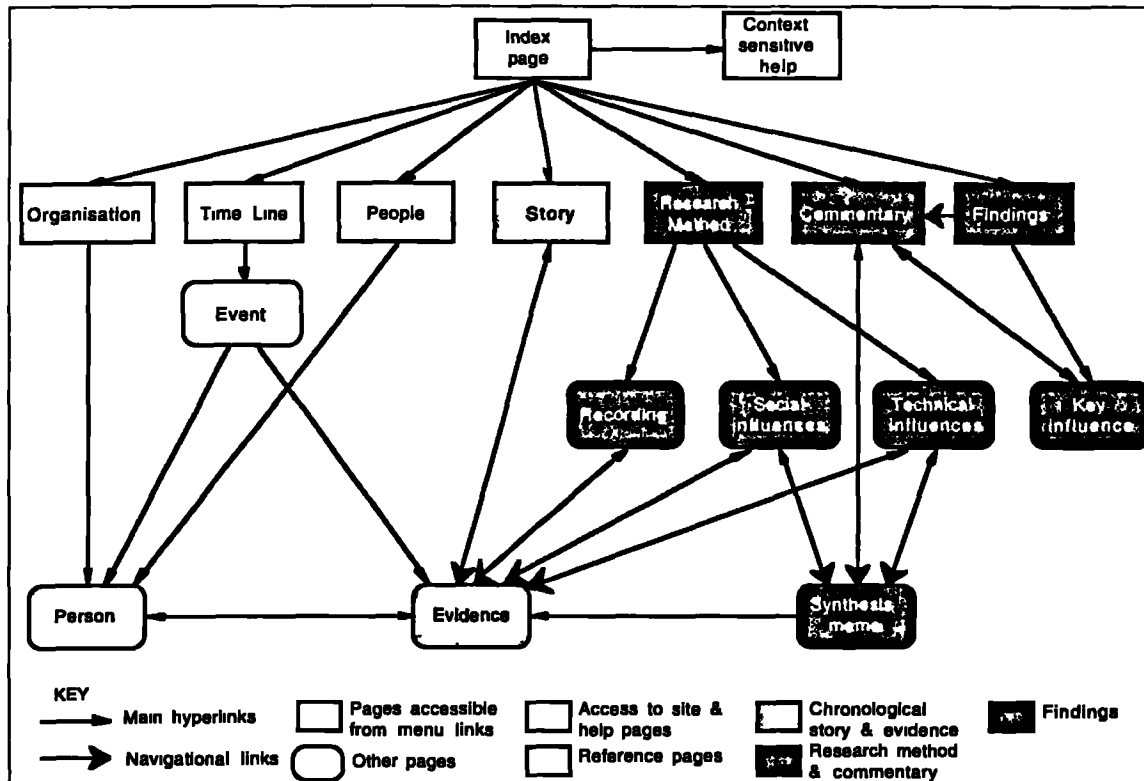


Figure 5.10 Structure of the web-site

Each box on this diagram represents page(s) in the web-site and the arrows show the main links. Menu links at the top and bottom of every page provide access to the main areas of the site. The reference pages contain information about the organisations, timings and people in the case (these are much extended versions of the information presented in chapter 6). The pages containing the chronological story (as presented in chapter 6) are linked to the underlying evidence from the conversation transcripts and case documents. The research method page summarises the method described in section 5.1 and allows the evidence to be accessed as it was recorded, or as categorised by social or technical influence. The synthesis memos on which the commentary was based, can be accessed from the commentary or from the social or technical influence that led to the memo. The findings page contains the table presented in section 7.4 linked to the discussion of each key influence (in section 8.1). The findings table allows the commentary to be navigated by each key influence.

The following sections illustrate how the web-site can be used to explore the case study in greater depth or to follow the route the author took to reach the findings.

5.3.3 Example use: exploring the chronological story in greater depth

The following example illustrates how the web-site may be used to explore the material presented in the thesis in greater depth. Figure 5.11 shows the 'Story' page in the web-site (accessible from the 'Story' menu link).

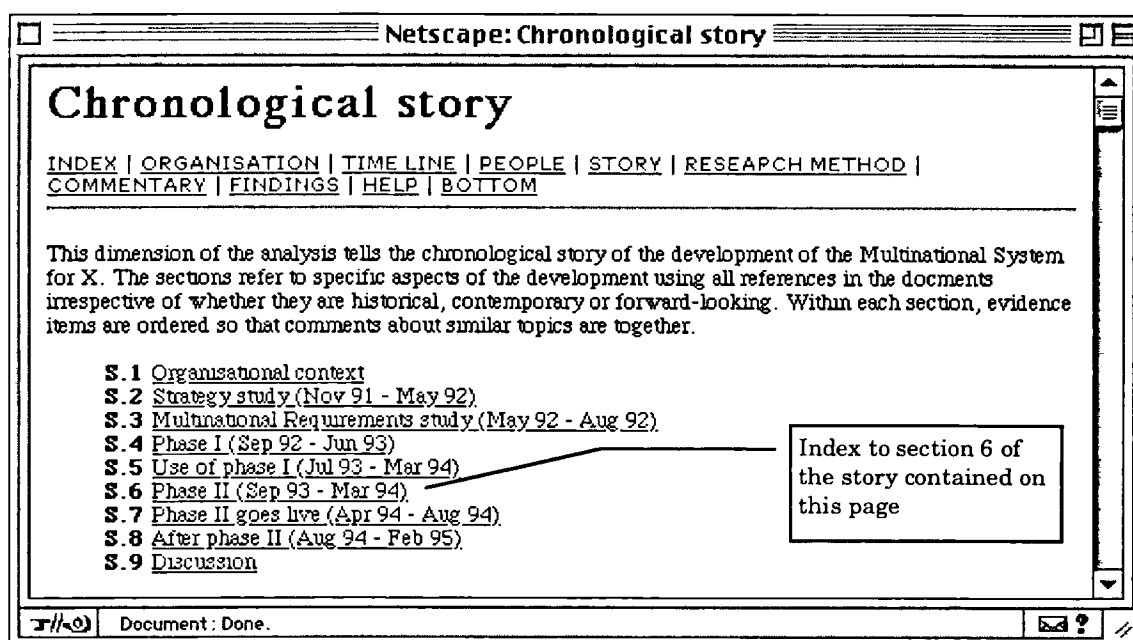


Figure 5.11 Story page contains the complete text of the thesis chapter 6

This page contains the complete text of chapter 6 but in the web-site version all references to people are hyperlinked to a page that summarises that person's role in the development. Following the link to section 6 and scrolling down shows section 6.6.5 in the thesis as it appears on the web-site as S.6.5 (see figure 5.12).

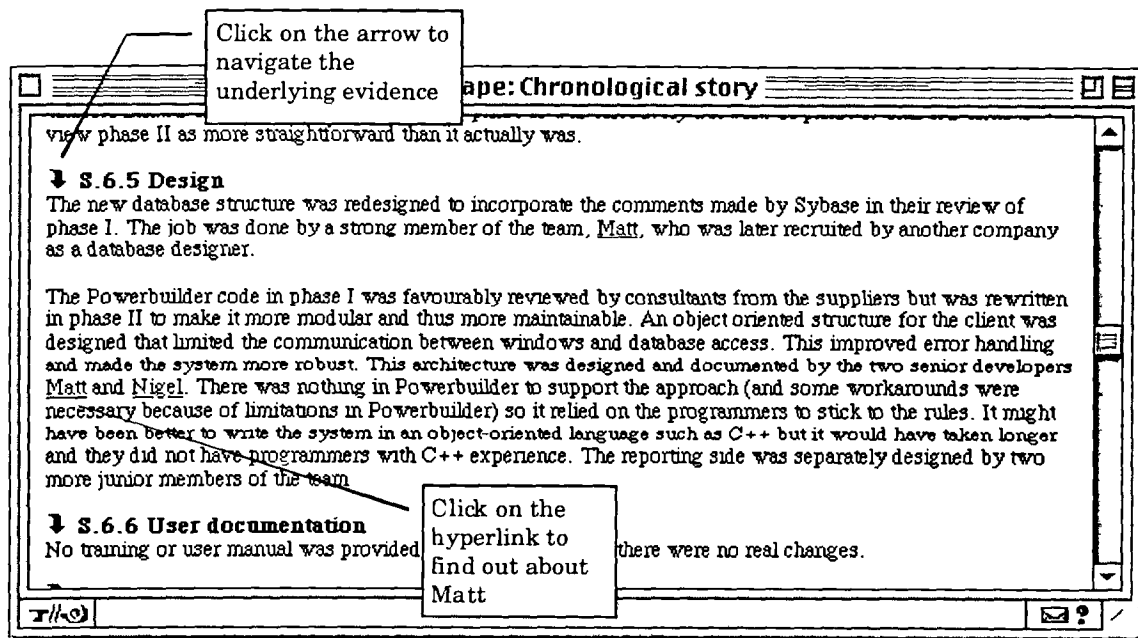


Figure 5.12 Section 6.6.5 of the thesis as shown on the web-site

For example, clicking on Matt shows a page providing information about Matt (see figure 5.13). Matt's page contains references to other people (hyperlinked to their pages) and references to extracts from transcripts or case documents that support the points being made (evidence extracts). The page also contains references to the original transcripts or case documents where Matt was an author or took part in a conversation.

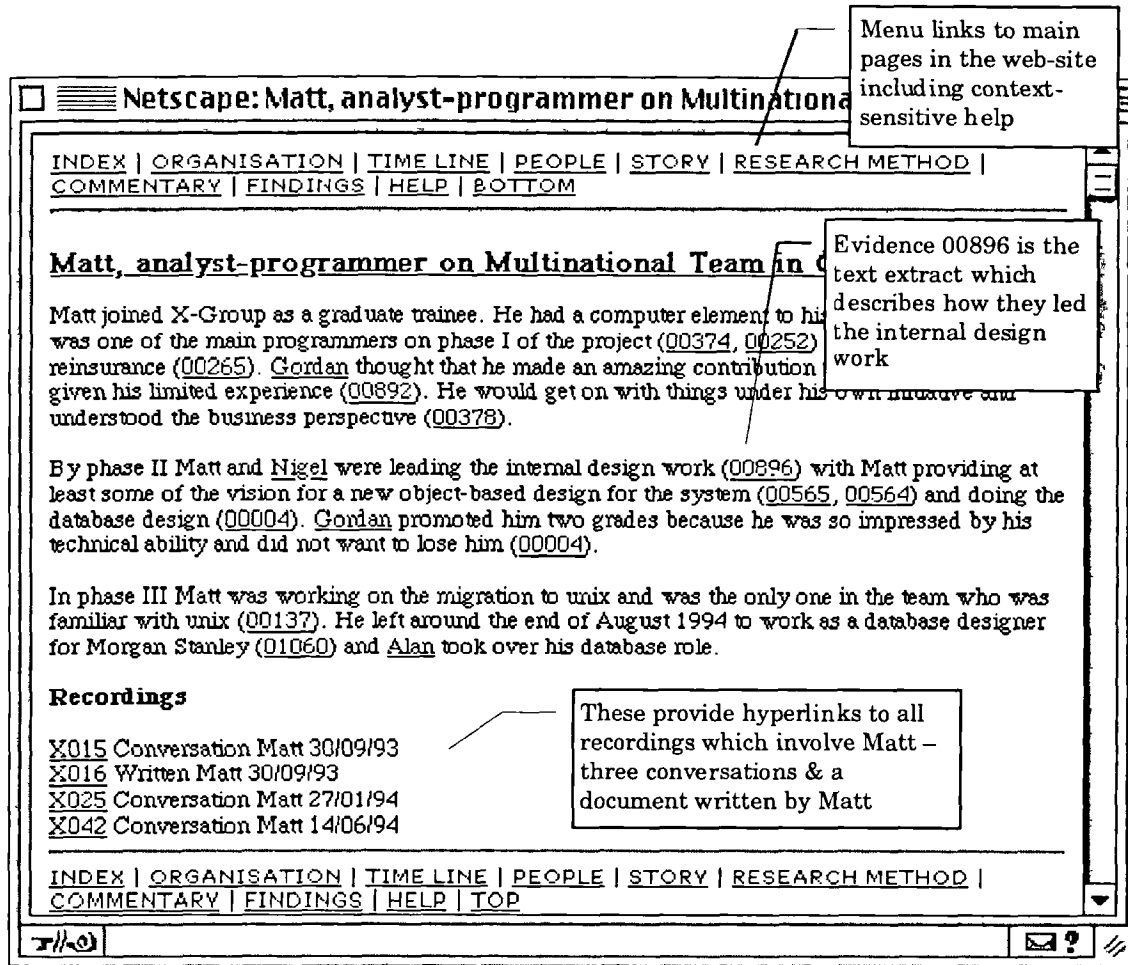


Figure 5.13 Person page for Matt

For example, to find out more about Matt's leadership on the internal design work, clicking on '00896' will show the page in figure 5.14. This is part of a conversation with Matt on 14 June 1994. The text shown is part of the full chronological story; text that was omitted as part of the construction (see section 5.1.2) is indicated by '...'.

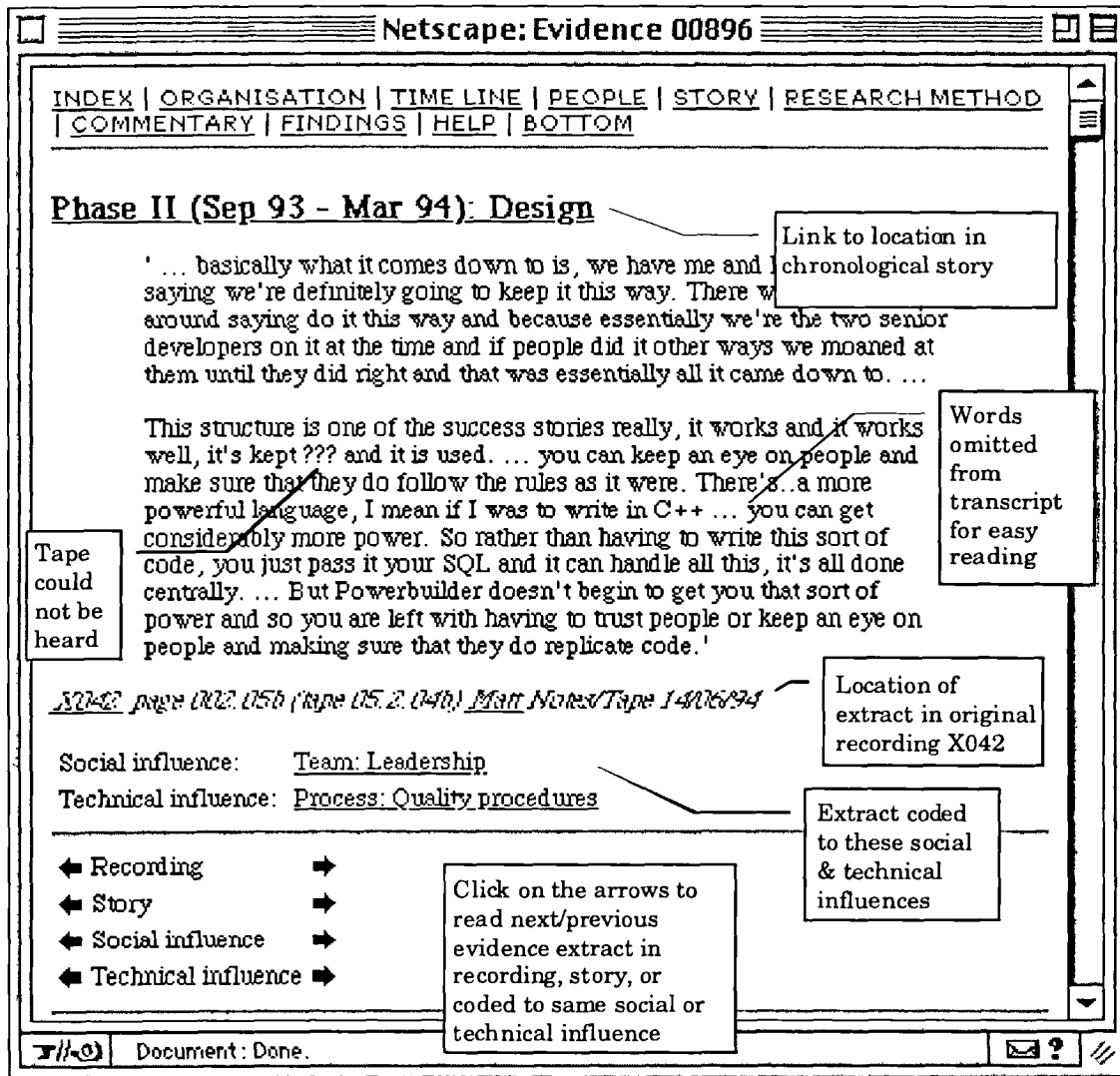


Figure 5.14 Page for evidence 00896, an extract from a conversation with Matt on 14/6/94

As figure 5.14 shows there are a number of ways in which the evidence can be investigated. For example, it is possible to see what Matt said next in the conversation (by following the 'Recording' hyperlink) or to see what else Matt and others said about the design of phase II (by following the 'Story' hyperlink).

At any point when reading the web-site version of chapter 6 it is possible to access (using the menu links at the top and bottom of each page) an organisational map ('Organisation' link) and a time line 'Time line' link). These are versions of figures 6.1 and 6.2 hyperlinked to further information about the parts of the organisation and events.

5.3.4 Example use: exploring the basis for a key influence

This example shows how the web-site may be used to trace how the findings were derived. The 'Findings' page (accessible from a menu link) shows where each key influence had an impact in the commentary (see figure 5.15). This page provides a hyperlink to each key influence and each section in the commentary and navigational links to the commentary events for each key influence.

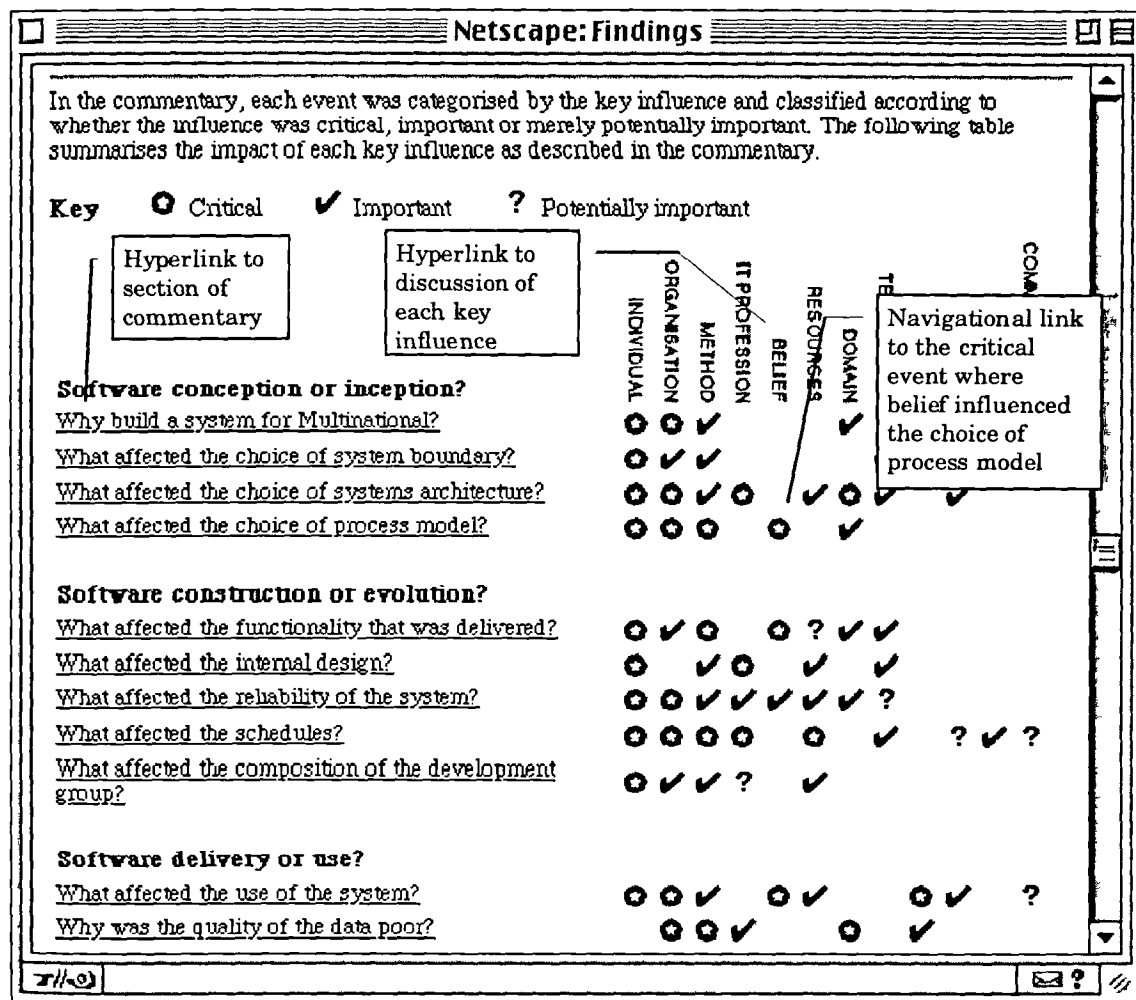


Figure 5.15 Findings web page

To investigate the rationale for the key influence 'Belief', clicking on the heading 'Belief' hyperlinks to the 'Belief' page (see figure 5.16) containing the discussion about the key influence belief in chapter 8 (section 8.1). This page contains hyperlinks to the 'Commentary' pages referenced in the text.

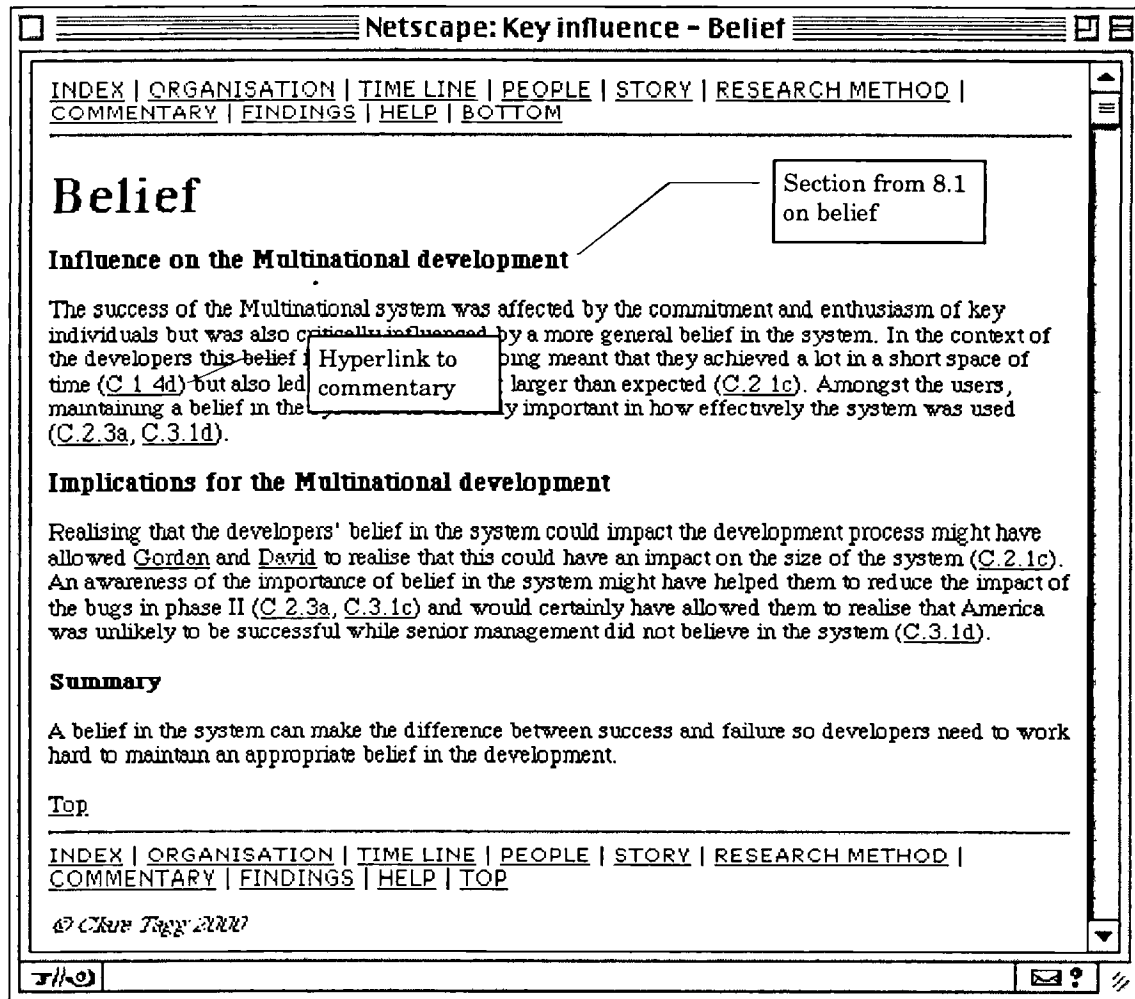


Figure 5.16 Page of key influence Belief from section 8.1

The effect of belief on the Multinational Development was written by reading all the events in the commentary coded to 'Belief'. These events may be viewed in sequence by clicking on the navigational link on the 'Findings' page for the critical influence of 'Belief' on 'What affected the choice of process model' (see figure 5.15). This displays the 'Commentary' page (see figure 5.17) positioned on the first event to be coded to 'Belief', 'Small and fast', in 'What affected the choice of process model'.

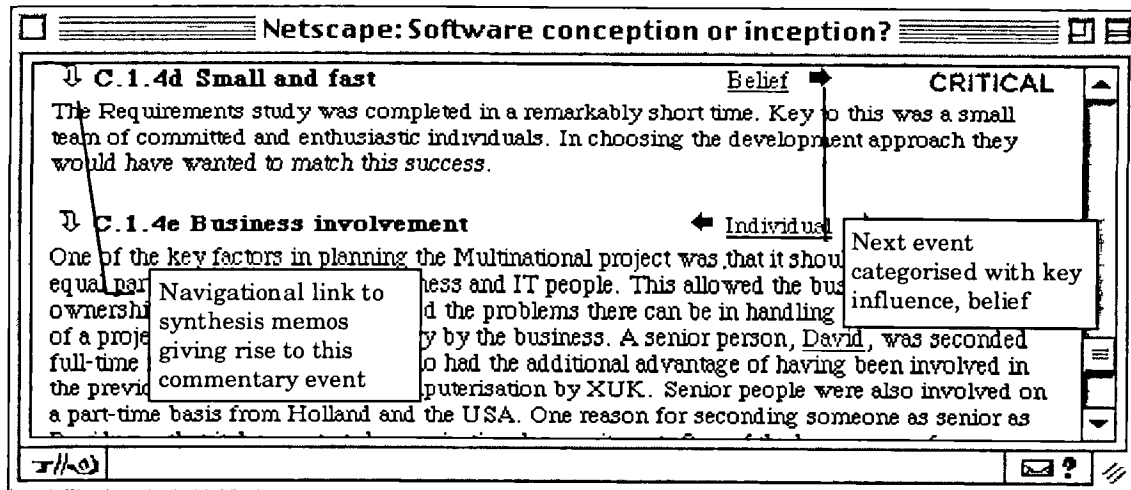


Figure 5.17 First event categorised to Belief in commentary page for 'Software conception or inception?', section 7.1.4

On the 'Commentary' page, the navigational links against the key influences allow all events coded to the same key influence to be viewed in sequence. So clicking on the arrow against 'Belief' will allow the reader to determine whether they would have drawn the same conclusions about 'Belief'.

To investigate more deeply, the reader needs to look at the origin of the commentary. The commentary was written to summarise the synthesis memos which referred to each event. There is a navigational link (see figure 5.17) to the synthesis memos that were used to write each event in the commentary. Figure 5.18 shows the result of following this navigational link for the event 'Small and Fast'.

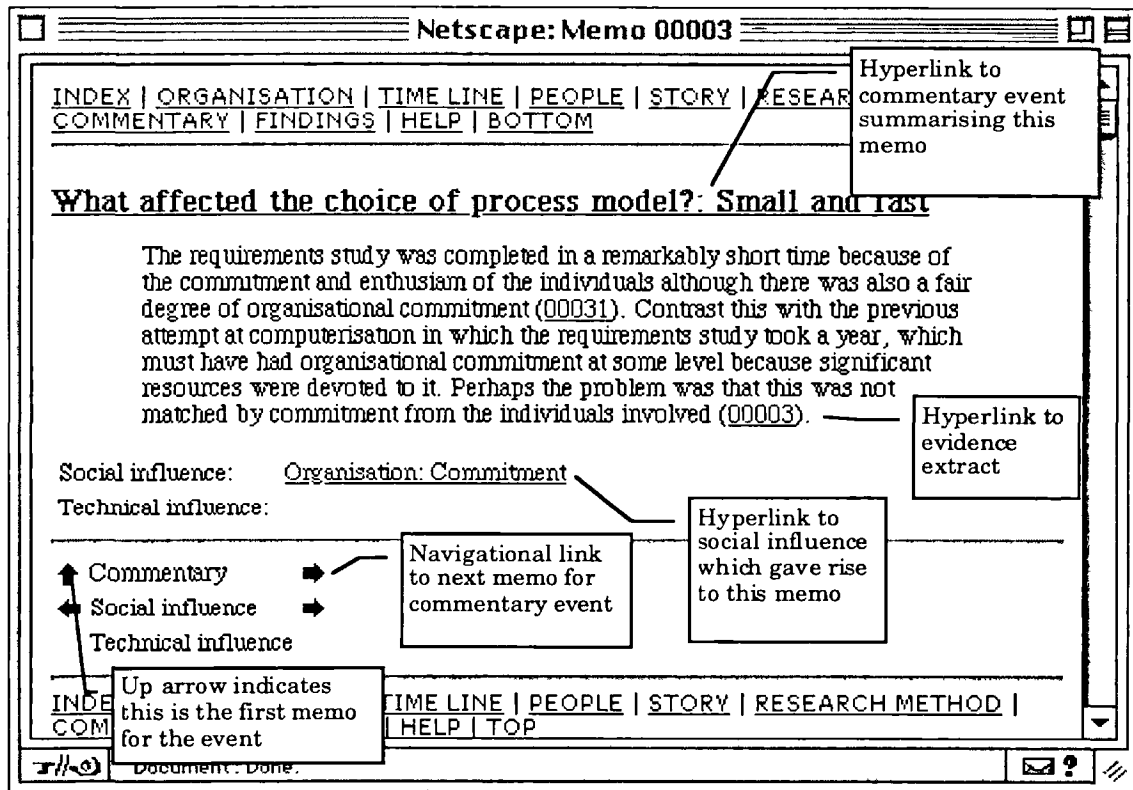


Figure 5.18 First synthesis memo for the commentary event 'Small and fast' affecting the choice of process model

The justification for this synthesis memo can be investigated by following the hyperlinks to the evidence extracts ('00031' and '00003' in figure 5.18). The synthesis memos were written by reading chronologically all aspects of the story coded to the same influence. In this example, the synthesis memo was written about the effects of the social influence 'Organisation: Commitment'. A summary of this influence can be found by clicking on the hyperlink (see figure 5.18) leading to the 'Social influences' page (see figure 5.19).

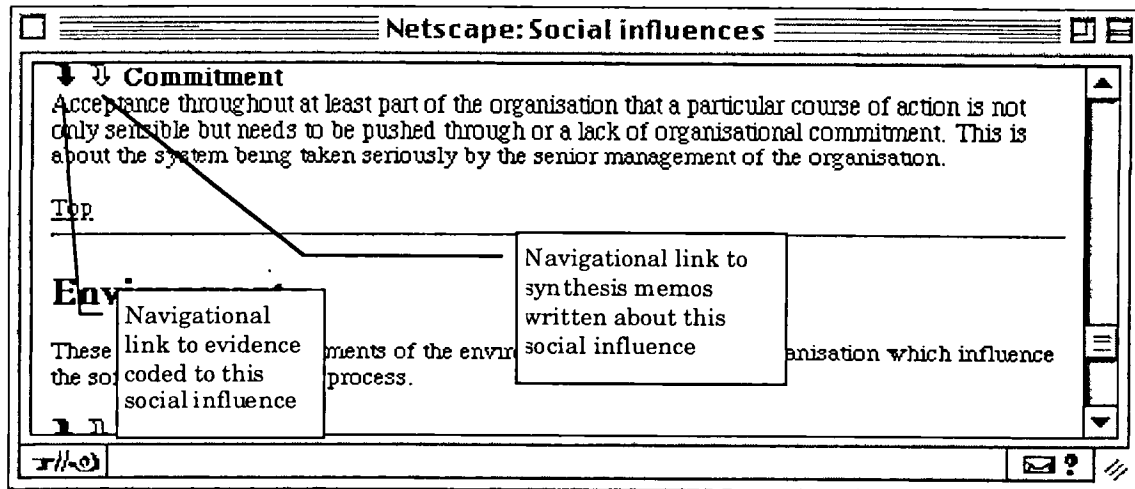


Figure 5.19 Page for social influences showing entry for 'Organisation: Commitment'

The evidence extracts coded to 'Organisation: Commitment' may be read chronologically by following the navigational link to the evidence (see figure 5.19). Figure 5.20 shows the first evidence extract for the social influence 'Organisation: Commitment'. Subsequent evidence extracts may be found by following the navigational link shown in figure 5.20.

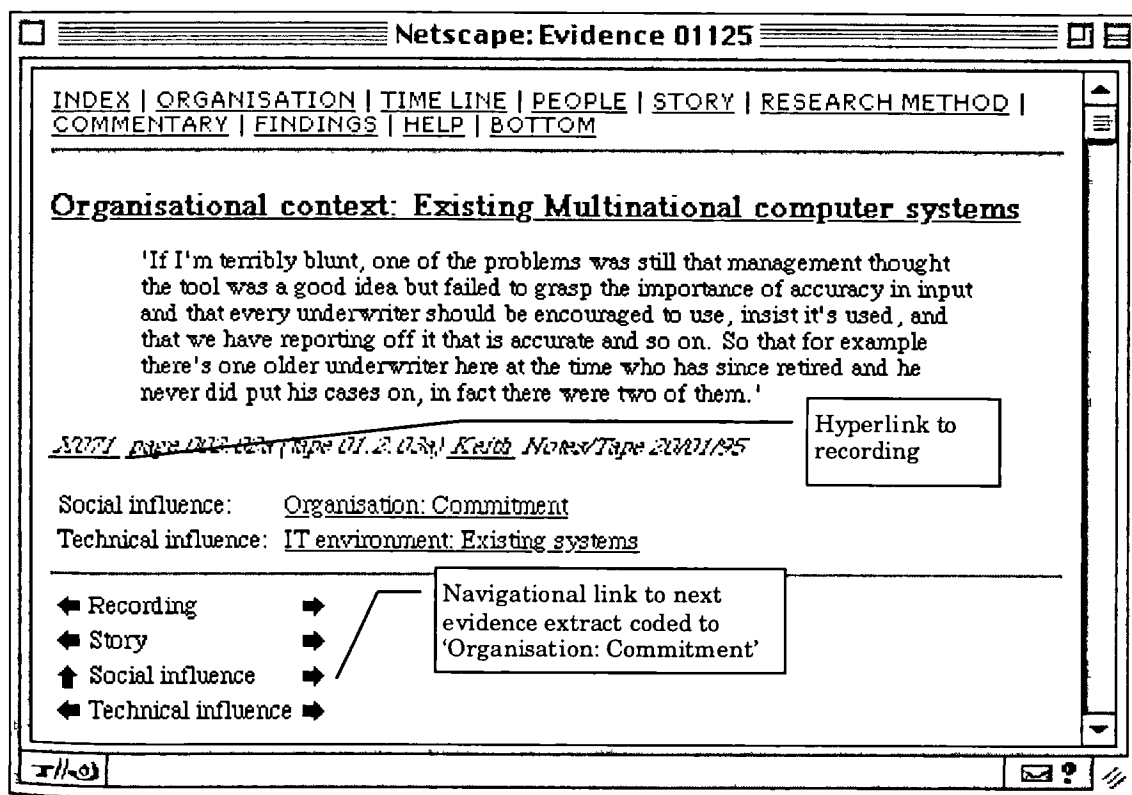


Figure 5.20 First evidence extract coded to the social influence 'Organisation: Commitment'

From the evidence extracts it is possible to go back to the original recording and following navigational links to read the evidence in the order they appeared in the recording.

However, it is not possible from the web-site to read the full transcripts of the conversations because the extracts exclude the text that was omitted during the development of the chronological story. For reasons of confidentiality and ease of development this text may only be read in the Hypercard version of the research.

This example has illustrated how the development of the key findings may be traced back to the original conversations and case documents. The 'Research method' menu link allows the development to be traced forwards from the chronological story, through categorisation of evidence by social and technical influences, writing of synthesis memos, writing of the commentary, categorisation of the commentary by key influences and the writing about the influence and impact of the key influences. Each of these steps is included in a hyperlinked diagram (see figure 5.21) which allows the reader to jump into a particular phase of the research.

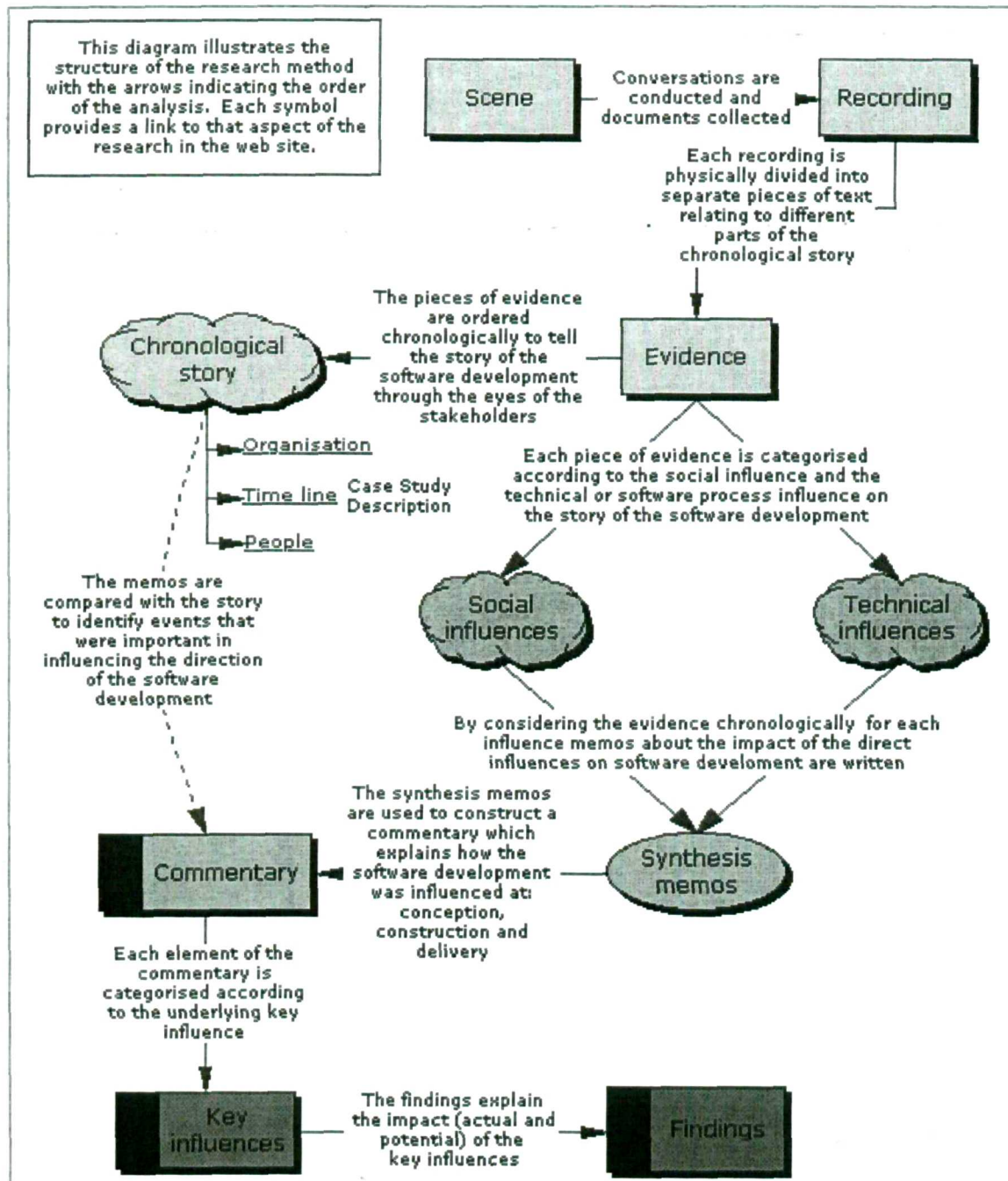


Figure 5.21 Hyperlinked diagram of the research method allowing access to each aspect of the research in the web-site

5.4 RESEARCH QUALITY

Chapter 3 established the requirements for quality within the research paradigm by redefining the traditional positivist terms of reliability, objectivity, validity and generalisability into concepts of goodness (consistent within a world-view and credible) and utility.

I achieve consistency in the research by repeatedly questioning my analyses and synthesis. Although the results are presented as finished products in this thesis, they arose from an iterative process that involved repeated reading and re-evaluation. In this, I was greatly assisted by the software (see section 5.2) which enabled me to look at the data from different perspectives. Throughout the research I tried to make my own position explicit and to discuss the research with the participants so as to challenge my world-view. Credibility of the research depends upon the reader believing the results. The detailed documentation of the development of the key influences provided in the CDROM allows the reader to confirm the credibility of the results (section 5.3.4 illustrates how this may be done).

This research is useful if the key influences are regarded as relevant to a wide range of software developments. Generalisation in qualitative research is often regarded as dependent on the user of the research recognising parallels with their own situation. To facilitate this the case is presented on the CDROM as a chronological story told in the stakeholder's own words (section 5.3.3 illustrates this). A story is used rather than the more recognised approach of well-formed argument because 'arguments convince one of truth, stories of their lifelikeness' (Bruner 1986 p11). Story-telling is the way humans make sense of things (Mishler 1986); it is a way of getting coherence (Nash 1990). The stakeholder's own words are used to increase the richness and credibility.

5.5 DISCUSSION

This chapter has outlined the data analysis approach used to identify the key influences on the software conception, construction and delivery in the longitudinal case X. The first stage of the method was to construct a chronological story of the case told in the words of the participants to reflect differing views of reality. As might be expected reality varied depending upon the speaker and whether they were looking forward, describing the current situation or reflecting with hindsight. The chronological story juxtaposes all comments relating to a specific event to highlight these differences. The construction of the chronological story effectively reduced the material under analysis without losing the context or richness of the data. The chronological story forms the basis of all subsequent analysis and is contained on the CDROM linked so that each contribution can be read within the context of the original conversation.

The following three chapters explore the chronological story and derive from the analysis of it. As discussed in section 5.1 this analysis involved three stages:

- description of the case including the organisational context and chronology of development - see chapter 6;
- synthesis of the technical and social dimensions to establish a commentary on the influences on development - see chapter 7;
- analysis of commentary to identify key influences on conception, construction and delivery - see chapter 8.

The research method described in this chapter provides an effective and organised way in which to explore a case study. The provision of the method on a CDROM using menu, navigation and hyper links allows the reader to follow the research in detail and thus improves research goodness and utility. The development of the method and CDROM would not have been possible without the Hypercard stacks developed to support the method. Despite advances in software to support qualitative analysis, there are no known commercially available software packages which provide all the features provided by these stacks.

6 Multinational system description

This is the first of the chapters that cover the analysis of the longitudinal case, X. X is a multinational insurance company that was formed in the X-Group in 1991. The longitudinal case covers the development of the Multinational system from late in 1991 to early in 1995. As explained in chapter 5, the complete analysis of the case is included on the CDROM. This includes hyperlinks to the original transcripts of conversations and system documents.

This chapter summarises the chronological story that forms the basis of the analysis of the case (see section 5.1.2 for an explanation of how this was constructed). The full text of the story is about 127,000 words so this summary is necessarily significantly abridged. The objective of the summary is to provide context so that the commentary and analysis of key influences in chapters 7 and 8 can be understood. One of the problems of the summary is that the original voices have been lost from the summary and with them some sense of the multiple realities. The CDROM contains the full chronological story with hyperlinks from the summary to the relevant parts of the story. The source of the very few direct quotes in the summary may be found on the CDROM. Each person has been given a pseudonym and details such as location have been changed to conceal the identity of the organisation. The glossary defines specialist terms used in the case.

Section 6.1 describes the organisational context of the development including the formation of X as a separate company. A diagram presents the main parts of X-Group referred to in the story along with the main players. The CDROM contains hyperlinks to each individual on the diagram and an index of all the people mentioned anywhere in the case.

A system for Multinational was included in the recommendations of an IT Strategy study of X undertaken in 1991/92 (see section 6.2). This led to a Requirements study (section 6.3)

and the start of the Multinational project. A diagram summarising the chronology of the development is included in section 6.3. The CDROM contains a more extensive chronology that is hyperlinked to the story itself.

The development was divided into three phases. The development of phase I is covered in section 6.4 with the use of phase I in section 6.5. Phase II development began shortly after phase I was installed and so proceeded in parallel with the use of phase I. Phase II development is described in section 6.6 and the delivery of phase II in section 6.7.

Following phase II there were a number of different developments including phase III and these are covered in section 6.8.

This chapter concludes with section 6.9, which discusses the conception, construction and delivery of the entire development.

6.1 ORGANISATIONAL CONTEXT

This section describes the organisational context of the development including the prevailing IT development cultures. The diagram below shows the organisational context of the case study; those parts of the organisation and people who were important in the story. Hierarchy within the organisation is not shown explicitly because people's positions changed during the three years and the diagram provides a summary of the whole period.

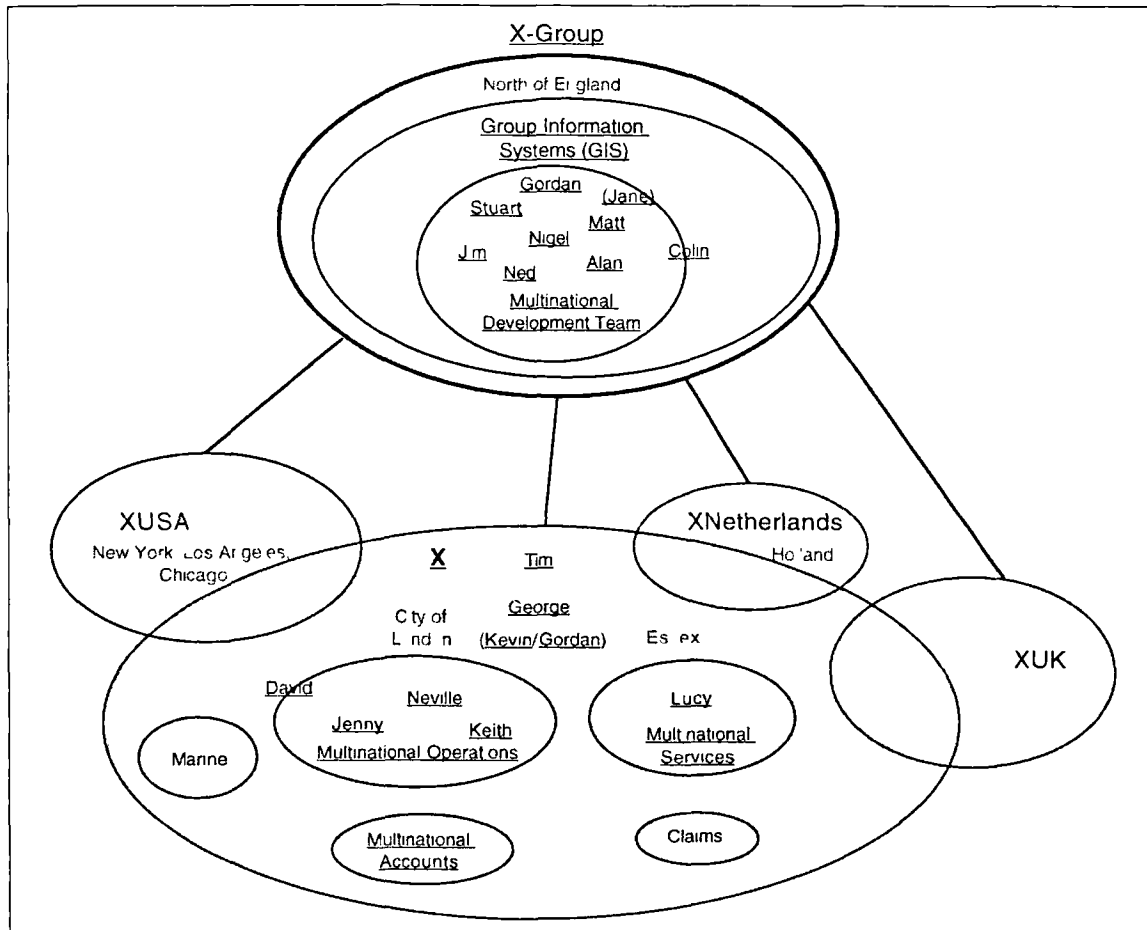


Figure 6.1 Organisational context, see appendix D.1 for a foldout version of this diagram with a summary of key players

6.1.1 History

X-Group is the parent company of a large international insurance group that writes almost all classes of insurance business. In 1991 a subsidiary, X, was created to bring together those parts of the group underwriting world-wide insurance risks for multinational companies. Initially, X specialised in providing a package of insurance services for multinational businesses. By 1993 the focus had been diluted as Marine business had been included to give the company critical mass.

6.1.2 Key factors for X

The market for large international insurance was well-defined with few major players. It

was a difficult time for the market when X was formed with companies competing on the basis of service. Service needed to be provided locally for handling premiums and claims but centrally for account management. X was created to provide a common identity and to give the business the opportunity to 'stand by its own merits' and so the focus was very much business-led.

6.1.3 Multinational Insurance

In Multinational Operations, underwriters worked with brokers acting for large multinational companies that wanted a programme of insurance covering a variety of risks (eg fire, employer's liability, public liability) for operations in many countries. The business was written by the underwriter arranging for policies to be issued in each country of operation. The premiums were collected locally and transferred back to the controlling office for the whole programme. Claims were mostly handled locally but the local office was reimbursed by the controlling office. Apart from currency differences, there were also local legislative requirements that needed to be met. Re-insurance allowed X-Group to spread the risk but required them to be able to aggregate exposures at a location.

The business was long term with renewal annually and a limited number of opportunities to quote. At the start of the development there were estimated to be 6000 multinational policies world-wide. Multinational companies were becoming more sophisticated in the way they managed their risks and thus more demanding as clients. The current systems in X could not handle these demands.

The nature of the business meant that data was often incomplete and distributed over a number of sites with no efficient methods of collation or aggregation. Existing systems and operational procedures were preventing X from achieving its potential or exercising an adequate level of control.

6.1.4 Physical organisation

The company headquarters of X were in the City of London with the biggest multinational

underwriting team. There were also teams underwriting in the States (New York, Los Angeles and Chicago), Holland and about 80 offices in other countries that were more involved in servicing the business. The Managing Director, Tim, was an aggressive Australian. Multinational Operations reported through the General Insurance Manager to Tim and was initially managed by David who had two Account Managers each managing a team of 10-15 underwriters.

Multinational programmes were serviced by a back office Services team in Essex, claims handled by a separate group in Essex and accounts in the City of London. Consolidation of X results was done at the Group head office in the North of England. All X-Group servicing was moved out of London to Essex in 1977 but Multinational Underwriting Services was only formed in 1992. Prior to this, servicing had been done by the underwriters. Lucy became section head for the Services team shortly after it was formed and at that time had four staff working for her. Although she reported for supervisory matters in Essex there were no other groups in Essex with similar working procedures because the XUK business was quite different.

6.1.5 IT culture in X

IT was perceived to be the way for X to compete by delivering accurate information, but in 1991 the competitors' systems were regarded as being old but better. When X was set up it decided to take services, including IT from local existing X-Group companies. In the UK, X got virtually all its IT support from XUK IT Department, including a computer and voice network, at a projected cost of £2.2 million in 1992. However the XUK mainframe systems were inappropriate for large case based multinational insurance which was quite different from XUK's business and X had little influence to change things. For these reasons, and because of the timescales necessary to get work done on the mainframe, a large number of PC systems had been developed by X. Some of these systems were critical to the business but had not been developed by IT professionals to conventional system development disciplines.

When X was created the main problem was a lack of complete accurate information to help

with underwriting control, customer reporting and management control. Although it was recognised that X would have to continue to depend on local operating companies for IT for some time, the difficulty of ensuring IT was aligned with business priorities was acknowledged. The problems of data availability and quality and the need to clarify work procedures were also identified early in 1992.

6.1.6 Existing Multinational computer systems

In Multinational Operations local mainframes were used for handling financial transactions and claims. These systems were generally not designed for multinational type business and did not hold the necessary level of detail. In the UK PC systems were used for Underwriting, Services, Overseas claims and Client reporting. In terms of computerisation Multinational Operations was far behind the rest of the X-Group operation mainly because of the one-off nature of their business but also because there was not the emphasis at management level on computerisation.

The first computerisation in Multinational was initiated by an underwriter using Lotus 1-2-3. Around 1990 a previous attempt at computerisation (by XUK and involving David) was stopped by Tim who argued that if X was not part of XUK they would not want to use the mainframe for Multinational. This was one of the reasons for initiating the IT Strategy study. When the XUK initiative was halted, one of the underwriters, Keith, who had a background of technical management in a joint venture in Jakarta but no experience of developing IT systems, suggested to David that he should make use of the work already done with XUK and develop a PC-based system for Multinational. Q&A was used because it was the standard tool in use on X-Group PCs at the time. Many of the underwriters used the Q&A system but it was seen as an underwriter tool and there was no pressure from underwriting management on the accuracy and completeness of the data. When Services was created a Q&A system was established in Essex with data transferred between the two networks at night. Later a Claims system was developed as well. The Q&A system was a single user database with separate files for each major class of business. The system produced basic operational documents and allowed the users to generate their own reports.

6.1.7 IT within X-Group

Although there were some communications between the IT departments within X-Group, each company largely operated independently. In the UK the two major operating companies were XLife and XUK and they both had big mainframe oriented IT departments processing huge volumes of data; XLife used PL1/IMS and XUK used Cobol/IMS. At the start of the Multinational system development separate initiatives for electronic communication were underway in the UK and North America and although the group had standardised on PC hardware (IBM PS/2s with at least a 386 chip), there was no standardisation on PC based tools.

6.1.8 Group Information Services (GIS)

In the UK, X drew IT services from XUK, XI and Information Services Department (ISD) of X-Group but in April 1992 the IT group in XI merged with ISD to form Group Information Services (GIS). GIS was formed to give priority to IT work for smaller operating companies. With these companies being smaller, GIS tended to specialise in PC local area networks (LANs) and package solutions but had inherited its system development methods from larger companies in the group. While there was some support for rapid development techniques, projects were expected to produce their own procedures for new tools or techniques. GIS had an internal Quality Assurance function.

6.2 STRATEGY STUDY (NOV 91 - MAY 92)

Tim commissioned GIS to work on an IT strategy study for X. The study was completed by Colin between November 1991 and May 1992. It looked at the position of IT within X and proposed a development programme for the next two to three years identifying Multinational as a priority area.

6.2.1 Conduct of the study

The IT Strategy examined X's overall business objectives for the next five years, assessed the current position of IT in X and set out what IT systems needed to be put in place to meet the business objectives. There was a view within X that the strategy was a necessary formality to justify the expenditure because with Tim as the new MD the systems were already seen as critical to getting the business going.

6.2.2 IT strategy findings

The IT Strategy proposed systems for Financial Consolidation and Reporting (an MIS), Multinational Operations and Non-marine Re-insurance which together would let X 'gain significant competitive advantage' and 'avoid significant mistakes'. It recommended a PC LAN based architecture linked to local mainframe systems with separate LANs exchanging data by diskette or file transfer. The use of a graphical user interface and iterative development were seen as key aspects of this architecture with packages being used wherever possible. It was recognised that there would be a need for local variations and that not all cases could be computerised but that development would be controlled at the centre. Moreover, as X was continuing to evolve, the IT systems would need to be flexible.

The emphasis in the IT Strategy, arising from a presentation that Colin did in October 1991, was of the new systems driving the local mainframe systems and progressively taking over the main business functions. It was felt that this would give X the necessary control over its IT but that it would require 'significant management effort and commitment' to achieve.

The Strategy highlighted the problem of incomplete data and the problems encountered in other parts of the group where senior professionals regarded data entry on a PC as a clerical task. It also identified that procedures would need to be clarified and common classifications (eg for territory and peril) would need to be used throughout X.

The Strategy proposed a plan of work for the next two years with a review at that point. After three years there were no plans to formally review the strategy although new work

plans had been developed.

6.3 MULTINATIONAL REQUIREMENTS STUDY (MAY 92 - AUG 92)

Following approval of the IT Strategy a joint X/GIS team undertook a three month IT requirements study for Multinational Business. The team was led by Gordan from GIS and David the Multinational Operations Manager. It also involved Keith the Multinational underwriter who wrote the Q&A system. It was an intensive project and there was a lot of commitment to it partly because X was a new company. The proposals of the Requirements study were approved by X management and development of the system began in September 1992.

The diagram below summarises the events in the development of the Multinational system. Each period of the development is described in more detail in the following sections. Timing of visits is included to show when the majority of the conversations occurred.

Year		Organisational events	Development events	Visits
1991		X created	Q&A system	
1992	Q1	MN Services created GIS created		
	Q2	David seconded to project Gordan leads project	IT Strategy Study	
	Q3	Lucy appointed to Services	Requirements Study	
	Q4			
1993	Q1			
	Q2	Jenny replaced David	Phase I installed	CU
	Q3	George appointed Technical team leader left		NEng
	Q4	Kevin appointed IT manager	Phase I review Marine project started	City
1994	Q1	Manny joins Marine team	Phase II installed	NEng
	Q2	Kevin resigned	Fin. Acc. sub-system started Unix migration agreed German standalone installed City requirements discussed	City Essex NEng
	Q3	Database architect left	Phase II review McDonalds system developed Client Services investigated	CU City
	Q4	Gordan & Karl replace Kevin Neville given business resp. Project manager recruited Collin appointed to Fin. Acc.	Data audit Unix installed in UK Process Automation invest.	NEng CU
1995	Q1		Unix installed in US Minor enhancements planned New standards developed New architecture planned US relaunch	City NEng

Key
CU Open University visit
NEng North of England visit
City City of London visit
Essex Essex visit

Figure 6.2 Time line of the Multinational systems development, see appendix D.2 for a foldout version of this diagram

6.3.1 Requirements study findings

During the Requirements study interviews were held with representatives of X in the UK, USA and Holland that resulted in three brainstorming sessions in the UK focussing on Underwriting, Claims and Finance. During this process, many of the separate requirements for Re-insurance and MIS identified in the IT Strategy study were incorporated into the Requirements study for the Multinational system. The IT Strategy

study suggested that the existing Q&A system could continue to be used, at least on an interim basis, for data entry but there seems to have been no consideration of this option in the Requirements study. As part of the work an entity model was produced.

Development was divided into two phases with phase I covering Underwriting and Claims for UK, USA and Holland and phase II covering Accounting functions. Phase I included a core database, automatic download and import of data from local mainframes, reporting and analysis of data and efficient exchange of data between X offices. Completion for phase I was scheduled for the beginning of May 1993 at a total cost of £785,000. This was in line financially with the IT Strategy study but the delivery date had already slipped by at least two months.

6.3.2 User involvement

A key feature of the project was the joint management of the project by David from Multinational Operations and Gordan from GIS. This partnership was seen as important by both parties and was made to work as an equal 'man and wife type' partnership with the business dictating what had to be done not the technology. The fulltime commitment of both a senior manager, David, and an underwriter, Keith, was regarded as a significant factor in the success of the project. Senior people were also seconded part-time to the project from Holland and the States but they did not work as closely with the team.

6.3.3 IT architecture

In parallel with the Requirements study an architecture study was undertaken to determine the platform, database, development tool and communications. It was intended that this would provide an architecture for IT within X. A decision was taken to use Unix servers for the larger sites but that development would proceed using a PC server whilst evaluation continued. In the event the power of PCs increased so fast that they were able to go live with PC servers in all locations. This saved a lot of money on hardware and so the project came in under budget. The LANs in London, Essex (2), North of England, Holland, New York, Los Angeles and Chicago were all linked together using ISDN.

6.3.4 Software

There was not an applicable off the shelf package available so the team evaluated a number of client server options. None of the relational databases had adequate Windows development tools and after an evaluation of the alternatives they chose Sybase for the database and PowerBuilder for the development tool.

6.3.5 Methodology

The team developed their own prototyping methodology with users and the development team working together to establish the system structure, validate the entity model and design the database, and, determine the functional requirements of Property underwriting, Casualty underwriting and Claims.

6.4 PHASE I (SEP 92 - JUN 93)

The development of phase I of the Multinational system, as proposed in the Requirements study, culminated in the installation at the end of May 1993 of Windows-based local area networks in each location together with the first part of the system in May 1993. The rest of phase I of the Multinational system was installed in June 1993.

6.4.1 Proposed functionality

Although the IT Strategy study had recommended that the Multinational system should drive the local mainframe systems it soon became apparent that this would not be feasible because it would not be acceptable to the IT people in the different companies. Instead, it was decided that the programme and policy information would be keyed into both the mainframe and the Multinational system. This did not involve much duplicated data because the mainframes held such limited information about Multinational policies and there were only a limited number of programmes involved (just over 500). On the other hand it was decided that claims would be downloaded from the mainframe systems into the

Multinational system, presumably because the volume of data made it worthwhile.

Data would be transferred around the Multinational wide area network (WAN) so that all information relating to a programme written in London would be in the London database and all information relating to policies issued by an office would be in that office's database. For offices without the database the information would be faxed to the appropriate office and entered into the system. So, for example, for a programme written in London with policies in New York and France, the policy details would be transmitted electronically between London and New York but by fax between London and France. Claims would follow the same routes with the policy details and claims from France being entered into the database in Essex.

6.4.2 Delivered functionality

Phase I concentrated on the data input and transfer aspects of Multinational with only a basic set of reports. The system consisted of modules for Quotations, Programmes, Policies and Claims. Acceptance of a quotation automatically generated a skeleton programme and confirmation of a programme generated policy records for each country. The system also included a basic ability to identify re-insurance parameters. The data input aspects of the system were much larger (737% was given by one source) than expected.

Phase I also included the installation of the LANs and WAN. PCs were installed on underwriter desks with standard Windows software and IBM terminal emulator software. Email systems were installed to conform to local standards. The users were impressed with the Windows technology.

The system was delivered in June (two months later than planned the previous September) and 3% under budget. This was regarded as a success because the system was so much larger than planned. However the system was very slow on the client PCs even though by the time they installed, the standard machine was a 486 33 MHz PC. Both Services and Underwriting were disappointed with the delivered functionality. In the case of Services there were things that the existing Q&A system did that the new system did not do and for

the underwriters things were designed but not implemented. The developers recognised that phase I was overly ambitious and that it might have been better to have left Claims out although there were benefits in including Claims in phase I. They also underestimated the work involved in installing it in the States but felt this was necessary for it to be seen as a global project.

6.4.3 Prototyping

A classic prototyping approach was used where requirements were defined in a workshop, converted into a prototype, presented and refined at subsequent workshops. Prototyping continued for some time because of the complexity of the business and because the team was not experienced enough in rapid development techniques to call a halt.

The prototypes were driven by the business model working first on Quotations, then Programmes and then Policies and Claims in parallel. Property and Casualty insurance was prototyped together but it was felt that it would have been better to develop the Property screens first and then extend them for Casualty. Although developing Claims and Policies in parallel was a bit messy it had the advantage of ensuring that the Claims system was considered from both a Claims and an Underwriting perspective.

Initially the workshops were attended by management with underwriters only being brought in at a later stage to check usability. The approach was very much based on starting from a blank sheet of paper, they did not analyse the way that the existing Q&A system was being used. Moreover they concentrated on the underwriters and did not really consider the Services area although Lucy was involved in most of the workshops.

The full day workshops started in September and continued every week or fortnight until February. Briefing papers were produced before each workshop setting out the aims of the workshop. Early workshops brainstormed with paper models but these were quickly followed up with PowerBuilder screens. Although there were only three or four major prototypes, each screen typically went through three or four versions. Technical debates were held outside the workshops.

Each workshop was attended by David, who generally chaired them, Keith, representatives from Multinational Operations, Services and sometimes Claims. A conference room in the City was wired so that the system could be demonstrated. The workshops were completely 'open floor' but sometimes did go round in circles because of changes in attendees. More was designed at the workshops than could be implemented in phase I.

The prototypes were coded directly in PowerBuilder, which the team found easy to use although it did crash on occasions. Once the prototype had been built it was installed in London and the underwriters were encouraged to put real case data in. The prototypes were also installed in New York and Holland to get their feedback.

Managers found it challenging being given a blank sheet and asked to define their systems but found it very helpful having a working system to comment upon. However it was necessary to manage expectations because development seemed so quick. In general the team found prototyping harder to manage because it was easy to get sidetracked by business issues and hard to say 'no' to changes particularly as they started with a very open attitude to the development and did not have clear boundary lines.

6.4.4 User involvement

The commitment of David, Multinational Business manager, to the project was important because he had the authority to make things happen and the background to see the impact of decisions. The management focus of attendees in the early workshops meant that sometimes the impact of decisions on underwriters and Services was lost. However, it is not clear that involving underwriters who were not particularly IT literate would have been very helpful. In a similar way Services staff sometimes found it hard to contribute to the workshops because they had a mainframe mindset and could not think about alternatives.

There was a view that the workshops gave the business ownership of the system but this does not seem to have been universally acknowledged, although the attitude to GIS was generally more positive than the attitude to XUK IT.

6.4.5 Design

The system was deliberately designed from a client perspective rather than the traditional policy approach of insurance systems. It was designed to support the underwriting process and to be easy to use with table selection wherever possible to improve data quality and speed input.

Mostly the system was developed directly from the prototyping. Re-insurance, which was highly complex and had never before been systemised, had to be rewritten. There were problems with the database design, which just grew out of the original entity model.

Although some rework was done in phase I, the database was scheduled for revision in phase II. There were also some inconsistencies in validation and messages. Although they wrote standards specific to the development approach, these were not particularly good and were not always adhered to.

6.4.6 Testing

Originally GIS planned to do unit testing and system testing. However the unit test phase was trimmed because there was no real consolidation period between the end of prototyping and live running. They also found it difficult to unit test in a traditional manner because the system was Windows based but system testing was carried out. Underwriters and Services were supposed to test the prototypes but this testing was limited because they were very busy. However, the system was tested extensively by David and Keith.

6.4.7 Installation

The networking in each location was a major element of installation for which insufficient time had been allowed. The installation of phase I was phased with the Windows technology introduced first along with the Client and Quote parts of the system. The rest of the system followed a month later but this meant that Services could not use the system initially and the rather slow, somewhat problematic start allowed some underwriters to

ignore the system.

A manual was not produced for the system because Tim thought the system was intuitive and there was online help. Unfortunately there was a bug in the online help system and America really wanted a manual.

6.4.8 Project management

There was a big push, with GIS working long hours, to get something in by the original deadline of May - although this was supposed to be the beginning of May not the end. The remainder of the system went in only one month later and under budget. The initial plans were formed on a sound basis and allowed for the inexperience of the team with the tools. Project controls such as weekly and steering group meetings were in place. However the plan was not redrawn when the system started to grow and the project team did not have the experience with prototyping to call a halt. The power of the programming language partly cushioned the team and they also worked very hard and cut corners with configuration management.

6.4.9 Team

In phase I the GIS team was led by Gordan who had two people working to him: Scott, his number two, was a technical team leader responsible for the networking aspects of the project and Stuart was the business analyst. Working under them were four or five analyst programmers including Elizabeth, Jim, Matt, Nigel and Ned. Elizabeth was one of the key analysts supporting Stuart but she left in March before the system went live. Gordan ran a flat, individually empowered team, which he felt was very strong with a good mix of skills. Stuart had a good understanding of the business, Matt and Nigel combined good programming skills with an understanding of the business, but Jim and Ned were both more technical and less good at seeing problems from a business perspective.

6.4.10 Quality assurance review

In November a post implementation review of phase I was undertaken by the Quality Assurance person with GIS. He recognised that the project had been successful in meeting budget, time and quality requirements. However he identified that the development approach could be improved in change control, database design, unit testing, technical documentation. He recommended that the approach to prototyping be changed so that it was only used for modelling the user interface. He also included a recommendation to appoint a team leader to manage programming in phase II but this was not included in the management summary. The team felt that his report was biased because he was 'steeped in traditional development methods'.

6.5 USE OF PHASE I (JUL 93 - MAR 94)

Phase I was used following its installation in June 1993 until the installation of phase II in March 1994. This section describes the conversion of data from the Q&A system and the subsequent use of the system by underwriters in the different countries, Services and Claims.

6.5.1 Organisational preparation

At the time of the Requirements study the resource implications of the new Multinational system had not been examined in depth but it was assumed that additional clerical staff might be required in Essex and that underwriter input could be absorbed within existing resources. It is not obvious that this was ever looked at in more depth or that system co-ordinators were appointed in the USA and Holland as recommended. There were some comments that the system was introduced at a particularly bad time for the underwriters when they were short staffed but that it was politically expedient to introduce the system on time because David was about to leave Multinational Operations. Once phase I had been installed they did have an operational review between October and December 1993 which took a broad look at information flow within the Multinational business in the UK

and made some minor recommendations for changes in the Multinational system.

The job of getting the system used by underwriters fell to Jenny. She joined Multinational Operations as an underwriter manager (one level below David) on 1 June 1993 and suddenly found that she was taking over the IT liaison role from David on a part-time basis. Although she was concerned about the need to prove herself on the business side, Jenny generally liked the role of IT liaison.

In Services the need for additional staff was eventually successfully argued but took time because the local head of department had little involvement in the unit as it was different from the other units in Essex. Jenny's background in servicing cases and the XUK discipline helped the Services unit to put forward their point of view.

6.5.2 Phase I conversion

In the UK and Holland data in the existing Q&A system was converted and downloaded into the new Multinational system. The idea was that underwriters and Services people would go into each programme and complete the fields that were not in the Q&A system. The conversion process took all of June 1993; for the first two weeks Services were updating the Q&A system and preparing the data for download, this was followed by two weeks when the data was converted and the system installed. The outcome of this was that Services started using the system with a backlog of work as they were without machines for at least two weeks.

About 85% of the data was downloaded but the quality of the data was doubtful because of problems with conversion and the poor quality of some of the data in the Q&A system. Conversion problems only affected some of the programmes so it was not until they were updated that the problems were found. One of the frustrations was that validation rules on the new system prevented some of the errors from being corrected by the users. The problems with data impacted on the image of the system and provided underwriters with an excuse for not using the system.

The UK decided to check the programmes on a rolling basis when they came up for

renewal. Although this would take a year to complete they did not have the time to update the data in one hit at the start. Even though systems were clearly 'flavour of the month', Jenny still had to persuade the underwriters to do the updating and in this she was not always supported by the other managers whose focus was on getting the business.

6.5.3 Use of phase I by underwriters

The official view of phase I was that it was a success as a system, a development strategy and as a new PC based architecture. However from an underwriter perspective it offered little, was not particularly convenient to use and it did not work well enough. In part this was because the system encouraged underwriters to be more systematic and culturally they were not interested in doing this.

In the UK it was decided that the underwriters would be responsible for checking the accuracy of the data at programme level with Services responsible for policy level data. The aim of having underwriters involved in data entry was to ensure they had ownership of their cases in the system and because they were best placed to enter programme details. Although new programmes were quite time-consuming to enter (one to three days was quoted), most work was renewals which was much simpler. Although this was a change from the existing situation where Services input all details to Q&A, it was not long since underwriters had input all details because Services was only formed the previous year.

The underwriters varied in their enthusiasm for using the computers; some believed in keeping control of their cases while others did not see the systems side as part of their job. This attitude did not seem to be related to computer expertise; in fact it was often the younger underwriters who were most resistant and their attitude was described by their manager as a 'cultural thing'. In addition, the underwriters only had one machine between two. One of the consequences of the rolling programme of conversion was that it could be several months between training and underwriters using the system in anger. There was no manual but they did have a refresher workshop in January highlighting system shortcuts.

Much of the process of case handling provided by the system supported Services and Accounts rather than the underwriters. In phase I the system was seen as a burden by many underwriters but Jenny likened it to a baby - it gives very little but when you see the first smile you know it was all worthwhile. Once the data was up to date, the system would provide management information so that underwriters would no longer be asked to provide exposure information. However, an ongoing problem was knowing what level of information to store about location data in order to provide the necessary detail on exposures.

Three months after the system had gone live David thought that Holland were very enthusiastic about it and had checked that all their policies had been correctly converted. He thought that the US were making good progress on using the system but had further to go. In practice, three months later, when Gordan visited the US for a management review, he reported that the Claims download was not operational in the US because of missing policies. Amongst the developers there was a view that 'the further west you are the more IT illiterate you are' so that while the Dutch embraced the IT ideals and there was some enthusiasm amongst underwriters in London, the US did not even know how to use a mouse.

Gordan was disappointed that there was resistance to the system given the use of prototyping and put this down to an innate dislike of change. However nine months after delivery there were signs of increasing enthusiasm.

6.5.4 Use of phase I by Services

When phase I was initially installed it was of no use to Services because it only allowed clients and quotes to be input. Throughout phase I, Services continued to use the Q&A system to service programmes which had not yet come up for renewal. One of the features of this kind of insurance is that there can be long delays in getting all the information necessary to process a case. This use of two systems meant that they had a very heavy workload and because they were full-time users of the system they suffered more from the crashes. Services also felt that phase I was inferior to the Q&A system which it was

replacing because it was slow and long-winded. Multinational Claims did try to use phase I but was not successful, in part because of missing policy data.

6.5.5 Phase I system problems

Phase I suffered from data inaccuracies, system crashes, errors in calculations and reports for overseas territories, and, was very slow. One particularly visible error was that help screens did not display properly. There were database crashes caused by a problem with Sybase that tended to affect Services more than London because they were accessing the database through a router. As this was a more complex system than Q&A it was more difficult to diagnose problems. Throughout the operation of phase I it was necessary for Keith to smooth things over to improve the image of the system.

6.5.6 Phase I data quality

Throughout phase I the quality of data in the system was poor. This arose because of a combination of problems with data conversion from phase I and a lack of commitment from underwriters to get the data right. In the UK this was partly because of the decision to wait until the programme came up for renewal before checking the conversion, so that it would be at least a year after installation before all the data was accurate or before management reporting was useful. So, for example, early in 1994 when Gordan was in the US to try and encourage greater data accuracy, there was an earthquake in Los Angeles but the system could not report X's exposure. In addition, not all underwriters entered their new programmes in the system.

The development team realised at this stage (six months into phase I) that they should have put more effort into ensuring data quality and actually built some time into their plans for phase II, but they still saw it as the users' problem. There was a view that there was a loss of continuity when David left that contributed to the data quality problem.

Actually Jenny was temperamentally better suited, with her Services background, to sorting out data, but she did not have David's authority. Although there was on the spot help from Keith on how to use the system and what data to enter, there was no detailed

monitoring of data quality throughout phase I and a certain lack of management commitment (excluding Jenny). This was a long standing problem that was inherited from the Q&A system and was one of the reasons why conversion was not useful. In retrospect it would probably have been better not to have converted the data from Q&A, then it would have been more obvious what was on the system, and there would have been fewer problems with data quality.

There was some discussion about using data entry clerks to enter data but the consensus was that this would cause more problems because it was not just a matter of entering the data, it needed to be interpreted correctly. One of the problems in America was that some fields were interpreted differently.

6.6 PHASE II (SEP 93 - MAR 94)

Originally phase II was divided into two parts: the first part, which was scheduled to go live in February 1994, was largely concerned with re-engineering the phase I system and eventually became known as phase II. The second part was concerned with adding Financial Accounting capabilities to the Multinational system, which became part of phase III.

6.6.1 Transition to phase II

There was a period of indecision following the installation of phase I while the business was deciding the priorities for phase II. This allowed the development team to focus on support which was useful but the project lost some momentum as a consequence. One of the reasons for the indecision was that the scope of X had broadened out; at this point the Marine project was added to Gordan's responsibilities. Work on phase II started in September but some urgent development work continued to be done on phase I right through until the end of the year.

6.6.2 Rationale for phase II

One of the main elements of the first stage of phase II was a re-engineering of the database and the PC client application to improve speed. This included upgrading the version of Powerbuilder. Phase II also included an extra policy class, security features and additional reports. One of the key limitations on the speed of phase I was the client PCs and the time taken to load screens. This was addressed in phase II by loading screens on system start up, so trading start up time for screen switching time. Phase I suffered from a lack of design arising from the prototyping process and phase II aimed to redress this by improving the design of the database and the application. The database was re-engineered to include referential integrity and the application was re-written in a modular fashion. The overall aim was to have an application that would be easier to maintain and extend.

6.6.3 Functionality in phase II

In practice there was very little change in functionality between phase I and phase II. The proposed enhancements to minor classes were dropped and a limited number of new control reports were included. The development of a suite of MIS reports and the installation of an adhoc reporting tool, Q+E were postponed. The view seems to have been that without the data, the MIS reports were not essential. The main change was that the whole system was effectively rewritten although with two exceptions it looked the same: the method of initiating renewal was deliberately changed and the order of policies was inadvertently changed. During this period work also continued on the import of claims data.

6.6.4 Development approach

As a consequence of the indecision from the business and the departure of Gordan's number two in September, detailed plans were not produced for phase II but the re-engineering was scheduled to go live at the end of February 1994. This was later revised to mid March to incorporate some specific reporting requirements. There was not quite the same pressure to deliver as with phase I but of course phase II was replacing an operational system so had to work properly from day one.

The development approach in phase II did not follow the same prototyping approach as in phase I and a number of technical design documents were produced during development. However, the development team underestimated the extent of re-engineering as phase II was effectively a rewrite of phase I. This led them to view phase II as more straightforward than it actually was.

6.6.5 Design

The new database structure was redesigned to incorporate the comments made by Sybase in their review of phase I. The job was done by a strong member of the team, Matt, who was later recruited by another company as a database designer.

The Powerbuilder code in phase I was favourably reviewed by consultants from the suppliers but was rewritten in phase II to make it more modular and thus more maintainable. An object oriented structure for the client was designed that limited the communication between windows and database access. This improved error handling and made the system more robust. This architecture was designed and documented by the two senior developers Matt and Nigel. There was nothing in Powerbuilder to support the approach (and some workarounds were necessary because of limitations in Powerbuilder) so it relied on the programmers to stick to the rules. It might have been better to write the system in an object-oriented language such as C++ but it would have taken longer and they did not have programmers with C++ experience. The reporting side was separately designed by two more junior members of the team

6.6.6 User documentation

No training or user manual was provided for phase II because there were no real changes.

6.6.7 Hardware problems

During the development of phase II there was a major problem with the use of Sybase in phase I. PCs accessing the database remotely over a router (those in Essex, Los Angeles

and Chicago) began frequently to lose their connection. This impacted on the phase II work as the team spent two weeks putting a trap in to handle the error gracefully. Sybase initially washed their hands of the problem but, after Gordan threatened to convert the system to Oracle, located the problem in the Novell communications protocol. Converting the system to a Unix communications protocol at a cost of £10,000 seemed to solve the problem. This reflected the fragility of the PC/Novell version of Sybase that eventually led to the replacement of the larger servers with Unix machines late in 1994.

6.6.8 Phase II testing

Phase II was subjected to unit and system testing but there was no acceptance testing because the change in functionality was so small. System testing also included some further unit testing and only revealed minor errors.

6.6.9 Installation plans for phase II

Installation of phase II involved converting the database so the changeover had to happen over a weekend. Stuart drew up a plan for the installation period and they planned to have someone in America and Holland when it went live. There was no fall-back plan for what to do if installation failed. Installing new versions of the software was much simpler as they had a utility that checked when a user logged on that they were running the most up to date version and upgraded them if they were not.

6.6.10 Team

At the start of phase II a Finance Director was appointed in X, George, who took over overall responsibility for IT from Tim. George was a qualified accountant who had been a consultant involved in projects with a high IT content. At the beginning of 1994 (two months before phase II was due to go live), George appointed another accountant, Kevin, as IT manager for X. Kevin had little experience of IT and although it gave Gordan a focal point for communication with X, it also absorbed a lot of his time and made it difficult to

resolve complex technical decisions. While Gordan was reporting progress to Kevin he was still reporting to GIS on resourcing matters.

Under George's new structure, Gordan was also given a more consultative role distancing him from a day to day involvement with the Multinational project. This only really formalised what had already happened because Gordan had been too involved in the Marine project in the latter part of 1993 to have time for the day to day management of Multinational. At this point the Marine project was taking off and Gordan diverted Scott's replacement, Manny to run the Marine project and broadened Stuart's role to lead the Multinational project including the phase II installation. He was concerned at the time that Stuart did not necessarily have the managerial or technical skills but aimed to continue having a close involvement himself. At the time Gordan was fully aware of the importance of project management in IT project success.

Stuart saw the team as a happy bunch of intelligent people and his new job to advise them and provide a buffer between the development team and the business. At this point the strong technical people on the team were Matt and Nigel. In fact Matt was regarded as so good, he was promoted two grades. Jim was still regarded as not good at seeing the business perspective and Gordan still did not feel he could completely trust Ned although in working with a new graduate, Alan, he was improving. Alan joined phase II of the project straight from completing a PhD in mathematics and after a week's Powerbuilder course started writing the reports with Ned. Each member of the team was given a lot of responsibility and contact with senior business people particularly when compared with the working conditions in other parts of X-Group.

Throughout phase II, business involvement was led by Jenny on a part-time basis but Keith continued to support the system in London on a full-time basis.

6.6.11 Project management

Phase II was originally supposed to have been led by Manny but his arrival was delayed by GIS. Gordan did raise the under-resourcing within GIS but it was not effectively raised

with X who were not necessarily aware that Gordan was spending a lot of his time working on the Marine project. The complexity of phase II was under-estimated by the team so that the under-resourcing was not viewed as that serious.

An overall plan for phase II was produced which had already been extended by two weeks before Stuart took over the coordination of phase II installation. When he took over he produced a detailed estimate of what was left to be completed and the deadline was extended by a further two weeks. There was limited monitoring and control throughout the development of phase II. Team meetings were held every one to two weeks but generally Gordan led by giving people responsibility and then letting them get on with it. There were regular meetings with the business people that involved some of the team members. When Stuart took over he recognised that more structure was required and aimed to have more detailed plans with short scale milestones. Unfortunately in drawing up the plans, Stuart had to rely on the judgement of the team members because he had not been involved in the development of phase II and did not have the technical knowledge to question their estimates. His revised plan did not allow enough time for testing.

6.6.12 Financial Accounts investigation

The other element to phase II was a Financial Accounting extension to the Multinational system. The analysis for this was undertaken during the Autumn of 1993 in parallel with the re-engineering of the database. At this stage the plan was to leave the Accounting in the UK and US with the mainframe systems but develop a Financial Accounting system based on Sun Accounts to run alongside the Multinational system in European countries. Sun Accounts had versions to run in each European country and was written in Sybase. At this stage the Financial Accounting sub-system was regarded as a relatively straightforward integration job. The plan was to develop it using a small team with a mixture of accountancy and technical expertise to run initially in Holland. The plans for this were presented at the end of January 1994.

6.6.13 Marine investigation

The Marine system was a separate development for X only interfacing with the Multinational system when reporting information about X as a whole. The analysis for this work was also undertaken by the team in the autumn of 1993 with the first presentation to the business in December 1993. A package solution was recommended for Marine and the choice of package was presented to the business on the same day as the Financial Accounting presentation at the end of January 1994. Marine was a big project (£2 million) involving a complex conversion and the management of an external supplier who was doing the tailoring work on the package. Contracts were due to be signed with the external supplier shortly after phase II went live.

6.6.14 Future plans

Around the end of February 1994, X led by the new Finance Director, George, started to think about longer term IT needs for the whole of X. While this did not directly affect the immediate developments on Multinational, the system was seen as the core of many of the future developments. Despite the negative comments, the business was proud of the system and saw its expansion as important to their future competitiveness.

6.7 PHASE II GOES LIVE (APR 94 - AUG 94)

When phase II went live it suffered from a large number of bugs that had a big impact on confidence in the system. Much of the effort in the four months following release was taken up with bug-fixing. An external review of the system was undertaken in August to learn from the phase II disaster.

6.7.1 Installation of phase II

The week before phase II was due to go live, Keith was asked to do some acceptance testing and he found about 27 immediate problems with phase II. As a consequence the business

users wanted to delay installation because they were aware of the importance to PR of a good launch to phase II. As most of these errors were relatively straightforward and many had already been found, Stuart wanted to continue with the release. He discussed the issue with Gordan who was on a short visit to America and they decided to go ahead in part because Gordan would then be in America when the system went live. The users raised the issue with Kevin but he was guided by Stuart and Gordan.

The day phase II went live the users were not necessarily aware of it being installed or of the changes. Phase II required the database to be converted and this led to some further data problems because of inadequacies in the phase I database design such as null entries in key fields.

6.7.2 Phase II installation bugs

People were unanimous in declaring the installation of phase II a disaster: there were 120 errors and queries in the first month, far more than they had with phase I. There were a large number of relatively minor errors, some serious data conversion errors, some unexpected changes in functionality and a Sybase problem that caused the system to hang. By the middle of June most of the bugs had been fixed but data problems continued to emerge. In particular records transferred to the States under phase I could not be accessed in phase II. The data problems caused a loss of faith in the system. There were also one or two cases where people used the conversion problems as an excuse for work that had not been done. The problem with Sybase was serious in that some servers just hung sporadically for no particular reason. The Sybase problem was not necessarily related to phase II but further sapped confidence in the system. The team did not have the low level access to the Sybase application to do anything about it and the Sybase suppliers could not fix it.

Although there were a large number of errors, the number reported included many enhancements and repeated reports of the same error. The team felt that the underwriters tended to over-react to the errors while Services were far more sanguine although they were far more dependent on the system. This was perceived to be a cultural thing

influenced by Jenny's attitude, which was that of a customer rather than co-developer. The over-reaction of the City was illustrated by a problem escalated to management level that turned out to be a LAN cable that had come unplugged. In contrast the underwriters in Holland were far more laid-back about the system problems. System problems were also used as a scapegoat so that management reactions tended to be far more extreme than the reactions from those using the system. In general Gordan protected the team from the worst fallout from the problems of phase II, although there were some comments that the developers were unnecessarily defensive in their reactions to the problems.

Shortly after phase II went live, Alan was given the job of managing the process of error reporting and correction. Errors were supposed to be reported through Keith using a standard form. Services found it difficult describing problems, which sometimes GIS could not recreate. They recognised the usefulness of a central point for recording errors but going through Keith just made communication more difficult. Services found it useful when they had a programmer on the spot but generally felt that the errors had been handled well. Some of the underwriters reported errors to Jenny who passed them through Kevin thus bypassing the normal reporting procedures. Jenny felt that the business users were not always given accurate information about error fixing although there were regular management meetings and lower level meetings to ensure the users were kept informed about the progress of bug fixes. In July 1994, GIS started a Multinational newsletter to aid communication with users. However during this period the frequency of the IT Steering Group meetings appears to have slackened off.

Alan's aim was to solve as many of the small errors as quickly as possible to minimise the number outstanding. He established a release process to ensure that fixing one problem did not introduce another problem. He also tried to make sure that things were tested properly before release although with the adhoc correction of data errors it was not easy to have a set procedure. One of the things that he discovered was that it was possible to be too helpful so that the users expected the team to solve data problems that it was more appropriate they should solve themselves using the system.

As far as the users were concerned the problems with phase II were extreme and continued

to emerge for the next six months. They felt that if only they had been allowed to do thorough acceptance testing, the problems would have been avoided. They also felt that some of the errors were so basic that they just showed a lack of care. However there was also a sense that part of the problems were caused by a lack of management commitment to properly resource the development team. There was some resentment at Gordan's involvement in the Marine project. Despite the problems, people I spoke to seemed to be committed to the system and kept telling me what a good system it would be.

The development team felt that they almost pulled it off but under-estimated the scale of the rewrite and hence the importance of user testing. The Sybase problem also clouded the issue. From their point of view it was disappointing that the technical improvements in phase II were ignored because of a large number of relatively minor bugs.

6.7.3 Phase II capabilities

Even without the errors the actual capabilities of phase II were not well regarded by the users. The speed of phase II in most areas was significantly improved but was still regarded by the users as too slow with some reports being particularly slow (15-20 minutes). The electronic transfer between countries was still not working fully. The number of locations used meant that re-insurance reports were inadequate and in general the reporting facilities were poor. The adhoc reporting capability of the Q&A system was due to be replaced by Q+E but this was not available in phase II and reporting was hampered by the lack of confidence in the quality of the data.

6.7.4 Views of phase II

Within the Multinational system in phase II there were 822 programmes (8925 policies) with 514 controlled in the UK, 149 in the US and 159 in Holland. Attitudes to phase II varied considerably between the different groups using the system although there seems to have been a general view that the system had potential but was not yet delivering that potential because of problems with data quality and a lack of functionality in the system.

Management in the City of London believed the system was good and would form the basis of ongoing developments that would improve their competitive position. Use of the system was just held up by the data quality problems.

There was a view that phase II was a public relations disaster with the City underwriters who were supposedly very critical of the system. It seemed to me that it was more that they were not particularly interested in making the system work and so were less likely to overlook system failures. Theoretically the system should have helped the underwriters to provide a better service to their clients but in practice the underwriters continued to work with their manual paper files in part because they needed the history. There was no standard form for these files and much of the information was not precise so that loading details of cases into the system was not straightforward. The problems with phase II allowed the data loading impetus to drift.

The system did make processing of cases easier particularly at renewal but my impression was that this was a relatively small part of an underwriter's job given that each underwriter was only responsible for between 9 and 20 cases. The other facilities on the PC, particularly email, seemed to be more useful to the underwriter than the Multinational system. So while there was a cultural antipathy towards using computers ('I was a highly paid typist', 'we don't like using PCs') this did not extend to stopping them using the system when it brought them real benefits ('the mail system is a brilliant tool ... it's replacing faxes'). In addition both underwriters I spoke to could see the potential of the system. The development team could not understand why, with so few cases each, the underwriters did not just get them in and right.

Despite the relative newness of the Multinational Services team there seemed to be a 'them and us' type relationship between the underwriters and the servicing team. This perhaps reinforced the underwriters' view that entering the data was not really their job although Lucy, Services' section head, reckoned that use of the system made the underwriters more sympathetic to what the Services team had had to put up with. As far as Services were concerned, phase II still failed to replace all the functionality that they had had in the Q&A system. Despite this, and the problems of phase II, Lucy at least was still very positive

about the system's long term use.

Despite some positive comments from the development team it was apparent that neither aspect of the Claims system were effectively used in phase II. The download of claims information from the various mainframe systems suffered from some problems with the data extracted and the data quality problems of the policy data in the Multinational system. The loading of claims data coming in manually did not appear to have happened. The main problem seems to have been a lack of interest in the system in the Claims department but I did not manage to talk to anyone from Claims.

The problem of getting the underwriters seriously to use the system in America continued into phase II. There were problems of differences in terminology between the UK and USA but perhaps more important there seems to have been a lack of senior management commitment to the system despite their involvement during prototyping.

Holland was reported to be generally happy with the system although they were affected by the same bugs as everybody else. When the system was installed on a standalone basis in Germany, training was provided by the Services Manager in Holland.

6.7.5 Post implementation review

In July and August 1994 a post implementation review of phase II was undertaken for X. This was a more thorough review than that undertaken following phase I because it was perceived that phase II had suffered far more problems on installation than phase I. It was conducted by Colin who had had no involvement with the Multinational development since the strategy study and so was seen as independent. He also interviewed me as part of the review to increase the independence of his report. The objective was to learn from the problems with phase II particularly with respect to GIS procedures and to seek views from the business on the benefits of the system. The review was not expected to look at phase I but as it found that many of the problems stemmed from phase I, it was extended to consider these issues.

The report found that the system was 'widely regarded as a good system, well conceived

and potentially offering significant benefits to X' with the potential to be important in X's future business success. Although the bugs were under control, there was a continuing lack of project leadership, the data load was incomplete and the system was not fully used. Despite the problems of phase II, GIS were regarded positively within X.

Project leadership of phase II was weaker than in phase I, although not non-existent (which is how Jenny interpreted the report). However, Gordan's reduced involvement did mean that the scale of the rewrite was not fully appreciated by all concerned. The report questioned the technical design of phase II arising from the lack of leadership but this criticism was not accepted by the development team. The report suggested that the development team were under considerable pressure from X senior management to deliver and that this may have led them to take short cuts with standards and procedures.

The report did highlight the need for the data to be complete for the system to be useful but this was not itemised and there was some confusion even six months later about the distinction between increasing the use of the system and improving data quality.

The report's main recommendations focused on the leadership issue by recommending the appointment of full-time business and IT project managers and increasing the role of the Steering Committee. The report recommended that a key task for these two project managers was to get the system into full use and get the data right. It suggested that X and GIS senior management should address the resourcing problems caused by lengthy reshuffles. It also recommended a number of technical checks.

The reaction of the directors to the report seems to have been fairly positive in that they saw it as a way forward. The development team felt that the report was unnecessarily negative. The good aspects of the system were not emphasized and some problems were reported out of proportion. Although most of the inaccuracies in the first draft had been addressed they still felt the report had an unfortunate slant. The report did recognise the role of the senior management in the problems of the system but its reception in GIS was influenced by the organisation's fear of failure. Despite their criticism of the report, GIS felt that the report 'focused the venom' and that subsequently there was a more positive attitude to the system.

Jenny was much happier with the report and felt it was fair. This was probably because, by Colin's own admission, the report had to satisfy Jenny and Neville. As a consequence of the report, she attributed most of the system problems to senior management in both X and GIS.

6.8 AFTER PHASE II (AUG 94 - FEB 95)

Following the post implementation review in August 1994, a number of different developments occurred many of them influenced by the recommendations of the review.

6.8.1 Unix migration

In June 1994 Gordan recommended migrating the large servers to Hewlett Packard hardware running Unix and a more advanced version of the Sybase database software at a total cost of £200,000. The main aim was to cure the Sybase crashes and improve reliability but this version of Sybase also offered Replication server. There was a view within the team (perhaps with the benefit of hindsight) that they should have gone for Unix initially as the PC/Novell combination was technically not appropriate for a database server.

The original plan was to migrate the UK server in mid August but the decision to go ahead was delayed because of a breakdown in communications between George and Gordan. In the end migration was scheduled for October 1994. The biggest unknown in the migration was the acquisition and set up of the hardware but they also allowed for five days Sybase consultancy in the budget because of the development team's inexperience with Unix and Replication server.

Matt, who designed the database for phase II planned the Unix migration but left the team at the beginning of September to go to a better paying job in New York. Alan, who had only joined the team a year earlier with no previous computing experience, took over and managed the Unix migration. He was assisted by another member of the team and some

new team members who were used for testing.

Migration to Unix was only planned for the large sites (UK, US and Holland) because the database problems only affected those servers supporting large numbers of users. In practice Holland with 20 users had not had problems but was migrated to Unix because of the Financial Accounting sub-system; Sun Accounts had had problems with the Novell version of Sybase. A Replication server running on the development machine would provide a warm standby machine. They also planned to use the Replication server for adhoc queries to protect performance on the live machine.

In theory migration was a simple task of running the scripts to build the database on the new machine with an identical structure and use a bulk copy program to copy the data from one database to the other. In practice there were some minor changes needed to the application and the original scripts did not match the live database but overall the migration was straightforward. Generally speaking Alan seems to have done well on the migration particularly given his inexperience but Gordan did report a communication problem with the support people which jeopardised the live date because a comms card was not ordered.

The Unix migration was very intensively tested with unit testing, system testing running the two systems in parallel and stress testing with the users running two systems in parallel. Gordan admitted that they had probably over compensated for the lack of testing on phase II and the testing did delay the release date.

Migration was generally trouble free although there were some security issues on day one. However, everyone had hoped that there would be performance improvements because the servers were so much more powerful, and in practice the migration did not make any real difference to performance. In consequence, performance remained a real issue and when the new project manager arrived, Jane, she blamed the database design despite all the work that was done on it in phase II.

6.8.2 Data audit

By August 1994 there were still quite a number of underwriters who were not up to date with loading their cases into the system. There were still problems emerging from the original download from Q&A which meant that those cases had not been looked at on the system for over a year. The teams were reorganising and the underwriters had agreed that cases could not be handed over until they were up to date in the system. With the proposed Financial Accounting sub-system it was essential that the data was completely accurate. A data audit was initiated from within X to assess the quality of the data in the Multinational system. Originally it was going to cover London, Holland and the US but America got a stay of execution because they did not have the data loaded.

The audit was undertaken largely by one of the underwriters I spoke to, assisted by Keith and Jim from the GIS team. The audit was managed by X's underwriting development manager reporting to the MD. The aim of the audit was to validate the paper records with the data in the system. It also aimed to look at how underwriters were using certain fields and identify training needs.

The first cases to be audited were Noah's, the underwriter doing the audit, and despite his computer literacy, this still revealed some errors in his data. The audit considered a 10% sample of each underwriter's cases and produced a report itemising the discrepancies. Most of the audit was undertaken by comparing paper records with print outs of the system but they did speak to each underwriter about their cases. It was a time-consuming exercise and could take a day to audit a single case because of the numbers of paper files involved. Jim investigated the reasons for discrepancies looking into the history of policies and identifying common mistakes. He also built a little system for recording the results of the audit in order to do trend analysis.

Each underwriter was given a report of their cases. A separate report was produced for management in UK and Holland and an overall report was produced. The reports did not compare the UK and Holland but in some aspects the UK was better and some, Holland was better. The report was sent to the directors but was not generally released and it was October before Neville got a copy of the report.

Overall the report found that the data completed by both Underwriters and Services showed a lack of accuracy and attention to detail. The report identified those screens and fields where people did not appear to understand what to insert, the number of transcription errors and those cases where it appeared just anything had been filled in. One of the underwriters claimed that most errors arose from the initial data loading period. As a consequence of the report a number of changes to the system were recommended including changing the layout of screens, tightening up on validation and some data quality reports.

6.8.3 Organisational changes

By mid 1994 X had established itself as a serious business and saw IT as a key to its development. By the autumn they were predicting significant increases in income from Multinational insurance without increases in staff. The company was restructured around the concept of profit groups with each group containing an IT person.

At the same time, Services were expanding with a view to them eventually taking over responsibility for entering all the policy details for renewals and new programmes. However, the expansion meant that they took on new people with no experience of Multinational insurance and it would be several months before they would have the necessary expertise.

In June 1994, Gordan's role was further refined although he was still working for GIS. He had the title of Systems Development Manager working to X's IT manager, Kevin. This meant that Gordan had a far wider role than just Multinational. However, Kevin resigned and although X initially sought to replace him they had difficulty recruiting. In the end they split Kevin's job into development and operations. In October, Gordan got the job of Development Projects and Standards Manager working directly to the Finance Director, George. He was now employed by X and spent about half his time in London. The advantage of this promotion was that he controlled the whole IT development budget for X including the overseas offices. He also no longer had to worry about the networks because that was part of the Operations Manager's job. When he took on the job, he did a

presentation to the Directors of X where they pledged their support and said they wanted to see all of X systems renewed or replaced. This included a plan to stop using the XUK system in a few years' time. Gordan was also responsible for hardware and development standards.

As part of the new IT management they decided that each project needed an IT manager, a business manager and an executive sponsor. Neville was given the business responsibility for Multinational. He was supposed to be working on it for 50% of his time and his job was on the line if he did not get the business issues sorted out. Although this had been decided in October, it was not announced until early January when his appointment was backed by the MD giving him authority to divert resources into getting the system running properly. Neville was the same level as David but Jenny also continued to be involved in the project.

In June 1994, in response to the problems with phase II, the directors of X decided that the Multinational project needed a clear project manager. Gordan agreed that Stuart was not up to the scale of this job as Multinational was rolled out over Europe. The need for a project manager was also echoed in the recommendations of the Post Implementation Review. In part this was because the scope of the Multinational project had grown and was seen as the core of a range of developments for X.

Normally recruitment of project leaders in GIS would be from within the organisation but in this case Gordan did not feel there was anybody internal who was up to the job. He recruited a contractor by approaching leading management consultancy organisations. He found that all the candidates he interviewed were strong in some aspect: good business knowledge, experience of the client-server architecture, experience of working overseas, but all were expensive. In the end he employed Jane from Touche Ross who was very strong on the methods side but had no previous experience of insurance. She started in late October so the recruitment process took about four months and the budget for her was 10% of the spend on the overall project. One of the reasons for employing an external person was that they would find it easier to be tough on X senior management and that seemed to be true.

Colin was also brought into the team to manage the Financial Accounting sub-system and other accounting elements of X systems. Although he was nominally working for Gordan

he was if anything senior to Gordan in GIS terms.

When I stopped observing the project the new management structure seemed to be working in that the project was re-energised.

The development team continued to be part of GIS even though Gordan was working for X, but it was no longer subject to the resource constraints of GIS. There was an ongoing discussion, supported by the Post Implementation Review, on the relocation of the development team to London or Essex because some felt that this would improve support and communications. Despite misgivings amongst the developers about the viability of such a move and mixed messages about communications, part of Jane's reorganisation of the team involved a small team being based in London. The team in the North of England also expanded with the intake of a number of consultants.

Keith remained involved in London but was sidelined in many of the new developments; he was no longer necessarily first port of call on the system for the underwriters with a development team based in London. He was not part of the management group controlling the Multinational development and his training role seemed to be diminishing.

6.8.4 Multinational phase III

Following Phase II, there were a variety of strands to the development of the core of the Multinational system which were loosely known as phase III.

Arising from phase II and the data audit there were many small enhancements to the system (and some errors continued to emerge). These included further work on re-insurance and tightening up on validation. In all there were over 150 requests and these were prioritised to be delivered in releases in March and June 1995. In addition further work was done on the system manual and documentation of the transfer mechanism.

The download of claims information continued to cause problems post phase II. Eventually in autumn 1994, Holland managed to establish a procedure for periodically downloading claims but in the UK there were still data problems that prevented some claims from

attaching to policies. Attempts to download claims in the US were abandoned because the quality of the data was so poor.

A reasonable number of standard reports were included in phase II but the users still felt they needed to produce their own reports as had been possible with the Q&A system. To enable them to do this, the team had planned to install the Q+E product initially in London followed by Services and Holland. Although Q+E was a standard product the team wanted to create some special views, test performance and do some training before releasing it. Initially, Q+E was scheduled for release in London in July 1994 but in practice it was not implemented in Holland until late in the year and had still not been implemented in London in January 1995. The delays in implementation of Q+E appeared to be due to difficulties over providing appropriate views and because the work was not given a high priority.

The Multinational system was installed in Germany on a standalone basis in July 1994 with the aim of linking them into the network once their cases had been properly loaded and they had moved offices. A large part of phase III, beginning in early 1995, was the rollout of the Multinational system to many other offices.

One of the most successful developments in Summer 1994 was a separate sub-system for McDonalds, X's biggest client. This system was based on a cut-down version of the Multinational system running on a standalone PC. It allowed claims information in five countries to be collected and emailed to Ireland for analysis by McDonalds. The system was easy to develop and showed how the system could be used for specific clients. One of the reasons the system was successful was that this was a new client and so there were no data quality problems as there were with the main Claims download.

During Summer 1994, Stuart worked with a number of X's other important clients to identify ways in which additional information (mainly claims related) could be collected and provided to clients.

In parallel with this, Jim produced a report on process automation. This looked at the gaps in the new system for Services; areas where they were still using other systems or IT could

be used to improve their efficiency. The main recommendations were for a diary, scheduling and control system, a fax/email gateway and the installation of Q+E. The developments were put on hold until the new year because the users did not want further automation until they were satisfied that the system was performing correctly.

One of the consequences of the problems of phase II was that the developers became much more cautious over the development approach. Initially, Multinational was developed using evolutionary prototyping but by the end of 1994 this approach was out of favour for all but small developments with prototyping being viewed as a technique for requirements capture in other cases. Of course one key difference was that they were much more familiar with the technology than when the Multinational system was started.

One of the developments of phase III was the introduction of new standards linked with the CASE tool already in use in other parts of GIS. Much of the work was done by external consultants but supported by Jane and Colin and following recommendations in the Post Implementation Review. One of the objectives was to reverse engineer the data so that they had at least a physical model of the database. The CASE tool was also used to develop a model of the design of the Multinational system. This was being developed gradually as enhancements were done to parts of the system. The CASE tool was used merely as a design tool, there were no plans to use it to produce PowerBuilder code. As a consequence of this work the development approach became much more traditional.

To support the roll-out plans to other countries the team were planning to use the Replication server capabilities of the Unix version of Sybase with a new server in Australia for locations in the Far East and Australia. In addition they were upgrading their communications technology from ISDN to frame relay. Sybase were providing consultancy advice on these new developments but Jane was also heavily involved.

6.8.5 Financial Accounting sub-system

When the Financial Accounting project team was formed in April 1994, the aim was to develop an integrated Multinational Financial Accounting system initially for Holland but

with a view to installation in other European countries. At this stage Gordan was aiming to make X independent of the XNetherlands system but the Finance Director clearly thought that X should be aiming to become independent of all mainframe systems. The Financial Accounting sub-system was seen as a relatively simple development that should be implemented by July 1994.

By June 1994 the developers had realised that the accounting side of Multinational was more complex than they had envisaged and had developed a mockup of the screens to use in a workshop to help with the writing of a formal specification. At this stage it seemed as if the complexity lay in specification not development because all the debtor/creditor aspects were to be provided by Sun Accounts; they were hoping to have a system in by the end of the year. During this period they began to consider using the accounting system eventually in the UK and so the UK started to monitor developments. In practice the business specification and projected timescales were presented in August. A two phase approach was proposed delivering the Sun General Ledger system in November 1994 and the integration with the Multinational system by June 1995. However the number and complexity of the screens to be added to the Multinational system was questioned by the UK and despite the early work it seemed as though much of the work could not be done in Sun Accounts.

At this point, in November 1994, Colin took over the project. Colin organised the project with a steering group involving both Holland and the UK and produced a plan, which showed implementation in Holland by November 1995 and this was only achieved by stripping the system down to the bare essentials. Colin produced a new specification in January 1995, which he asked the users (steering group and directors) to sign. One problem which took time to resolve was the links with the Multinational system because if underwriters changed fields once a policy was live it would be a 'nightmare' in accounting terms. One of the difficulties that Colin was going to have to manage was holding the requirements of the Accounting sub-system constant when the Multinational system was changing. Although Colin was derogatory about previous attempts at specification, the specification agreed in January still appeared to be high level and did not have an identifiable data model.

Colin was proposing to develop the Financial Accounting sub-system in a traditional way and had already produced a note outlining how the programming was to be organised which was similar to how things were organised when Colin joined as a trainee programmer in 1977. His argument for this approach was that accounting was a very precise problem so there was no room for prototyping although he did plan to keep the users involved with the development as much as possible.

Colin's plan was to produce a system at the outset that would be appropriate for the UK although it would be installed in Holland, France and Germany first. This meant that it would be the end of 1996 before the data had to be exactly right in the UK. Although it was expected that each country would need minor modifications there was a real sense that they were defining the accounting policy for X through the system.

6.8.6 Marine system

There continued to be no link between the Marine and Multinational systems except possibly in an overall information system. The system was using the same kind of technology as the Multinational system but was developed by a package supplier. Implementation started in June 1994 with a delivery date in 1995. It was a big project with some difficult conversion problems but Gordan was very confident in Manny who was leading the team.

6.8.7 Post audit data quality

Despite the findings of the data audit and the changes in management structure, data quality problems continued to bedevil the Multinational system. The 40% error ratio revealed by the audit and the continuing phase II problems meant that they missed the end of year deadline for getting all the data sorted out in the UK. There seemed to be mixed views amongst the users about whether Neville's appointment would solve the problems but the signs when I last visited the City (in January 1995) were not good: there were still some cases that had not been updated at all on the system, management were not using reports produced by the system to control data quality, little had been done

to improve attitudes to data quality following the data audit and changes in underwriting policy were not being reflected in changes in the system. One of the problems in getting the data accurate was the consistent handling of awkward cases within the system.

The Claims part of the system was only really working effectively in Holland although there were signs that the situation with Claims was improving in the UK. There seemed to be continuing problems over understanding how the electronic transfer of information was supposed to work and the Claims part of the system was still very slow. There were still old Q&A systems containing claims information prior to the start of the Multinational system which needed to be loaded into Multinational.

In January 1995 the decision was taken to relaunch the system in America because the data was so poor and the original underwriters had all left. The relaunch involved developing new training materials and a user manual and Jane and Jenny training each of the underwriters. A plan was agreed for loading the data in America.

In rolling out to Europe far more emphasis was placed on training than previously. Moreover, in February 1995, George charged the development team with managing the data quality initiatives.

6.8.8 Future

By February 1995 there was a clear view on the role of IT within X although the IT strategy had not been rewritten partly because they had 'gone out of fashion'. The role was far wider than Multinational but then X encompassed more than Multinational by 1995 and IT was seen both as a driving force and in a supportive role within the business. The new management structure of IT seemed to be promising a higher profile for Multinational and there were achievable plans for rolling it out to other countries. However, there was not clear evidence that the 'hearts and minds stuff' of X acting as one organisation and sharing information had really been solved.

6.9 DISCUSSION

This chapter has summarised the chronological story of the development of the Multinational system for X from its conception in 1991 through three phases of construction between 1992 and 1994 to its delivery from May 1993 onwards. It summarises the chronological story, constructed from conversations and documents as described in section 5.1.2. The full story is contained on the CDROM.

6.9.1 Conception of Multinational system

Officially the Multinational system arose from the recommendations of an IT Strategy study conducted for X in late 1991 and early 1992 by Group Information Systems (GIS). X had just been created as a separate organisation focussing on multinational insurance and needed to consolidate its world-wide activities which were using a variety of IT systems in the different companies in the X-Group. As the development of Multinational began, a group was formed in Essex to provide back office type servicing to the multinational underwriters.

The aim of the system as presented in the IT Strategy was for an information system to support multinational underwriting but the scope broadened during development to include operational aspects. By 1995, the Multinational system was regarded as the core of an IT system for X. The relationship between the Multinational system and the other systems used by X varied accordingly.

The IT Strategy and Requirements studies established an architecture for Multinational based on a client-server architecture running on PC LANs with Unix servers. The Sybase database server was used with the client application developed in PowerBuilder. The system was actually developed using PC servers, but when Sybase proved to be unreliable in this environment, had to be converted to Unix in late 1994.

6.9.2 Construction of Multinational system

An evolutionary prototyping approach was adopted for phase I. Part of this approach was a strong emphasis on the business and David, the manager of Multinational Operations and Gordan from GIS, jointly managed the project. This partnership was regarded as very successful. Phase I focussed on establishing the infrastructure in the first three countries to get the system (UK, Holland and USA) and developing the data entry screens. It turned out to be much larger than planned and there was a delay of about two months in its full installation.

There were flaws in the database design of phase I and the client application was very slow. One of the main tasks in phase II was to re-engineer the database. In practice the majority of the system was rewritten in phase II. In phase II analysis began on a Financial Accounting extension to the Multinational system. Initially this was regarded as a relatively minor development for Holland but the scope of the application grew to include the UK and its development time increased substantially. This became part of phase III along with a number of separate development initiatives covering adhoc reporting, client services including a sub-system for McDonalds, and further process automation.

Apart from Sybase problems, phase I had some errors but very few in comparison with phase II. Phase II had so many errors that the users lost confidence in the system. Most of the errors had been fixed within four months but this delayed the start of many of the phase III developments.

Phase I benefited from a concentration of senior management input but the team changed at the start of phase II. George, the Finance Director was appointed in X with responsibility for IT and he replaced the MD's involvement. David left Multinational Operations and Jenny who was less senior replaced his role in development. Gordan's number two, Scott, left and was not replaced and Gordan was allocated an additional large project for Marine. Gordan was further diverted when George appointed Kevin as X's IT manager. Some of the problems of phase II were attributed to this loss of leadership and it was eventually addressed in phase III. Gordan was promoted to have budgetary responsibility and had two new project managers working for him. Jenny's boss was made

responsible for the business aspects of the system with express authorisation from the MD.

6.9.3 Delivery of Multinational system

Data for phase I was converted in Holland and the UK from the existing PC system. Unfortunately this data was not very accurate and some errors were introduced by conversion. When phase II was installed the data was converted from phase I and a number of further errors introduced.

In the UK the plan was for underwriters to check and add additional data to programmes as they came up for renewal. Thus it would be a year from the installation of phase I before the UK data was updated and Services had to continue to process accounts that had not yet renewed on the old PC system. As a consequence of the problems of phase II and a reluctance amongst underwriters to use the system, the data had not been updated eighteen months after installation of phase I.

Holland had a smaller number of programmes and checked the converted details immediately. The underwriters were also rather more enthusiastic about using the system. However an audit of data in UK and Holland carried out fifteen months after installation revealed data inaccuracies and omissions in both UK and Dutch data.

In the USA the data had to be entered from scratch and the resistance to using the system was such that it had to be re-launched in the USA eighteen months after going live.

The problems with data quality impeded the smooth running of the Claims and electronic transfer systems. However there were also technical problems, problems of understanding and of ownership that had an impact.

6.9.4 Conclusion

Although there were technical failures in the development of the Multinational system, notably the database design of phase I and the errors in phase II, the development was generally undertaken in a professional manner. Despite this eighteen months after the

first phase was installed the data was still so poor that the system was not useful as an information system.

Chapter 7 presents a commentary on the conception, construction and delivery of the Multinational system developed as described in sections 5.1.2 and 5.1.3. This is analysed to identify the major influences on the development process that produced this dichotomy between the system's relatively successful development and ineffective use.

7 Multinational system commentary

Chapter 6 described the chronology of the Multinational development, summarising its successes and failures. This chapter unpacks the reasons behind the system conception, construction and delivery to make sense of those successes and failures.

The commentary presented in this chapter derives from a synthesis written by analysing the chronological story from a social and a technical perspective. As described in chapter 5 (section 5.1.3), memos were written highlighting the consequences of each direct social and technical influence. These memos were categorised according to their relevance to the conception, construction or delivery of the system. Development of this grounded categorisation resulted in a series of questions about the software development (eg 'Why build a system for Multinational?') and a series of events that had a bearing on the question (eg 'IT Strategy', 'New MD'). Discussion of these questions and the effects of the events forms the bulk of this chapter; this was written by considering the memos coded to each event. In the version of the commentary on the CDROM there are links to the memos and hence to the chronological story.

In writing the commentary the aim was to focus on the influential events and contrast the impact of the technical with that of the social. Hence the three over-riding questions of this chapter (sections 7.1, 7.2 and 7.3) are:

- Software conception or inception?
- Software construction or evolution?
- Software delivery or use?

There were many interlocking reasons why events unfolded as they did but the commentary concentrates on those events that were critical, important or potentially important. An event is classified as **critical** if it made a significant difference to the

software development. For example, if X had not been created as a separate company it is very unlikely that they would have built the same kind of system for Multinational.

The impact of an event is classified as **important** if it had a clear influence on the development process but changes in this area would not have resulted in a major change to the software development. For example the difficult state of the market when X was created meant that an IT system that could help establish their position was important but the system would probably still have gone ahead if the market had been easier.

The impact of an event is classified as **potentially important** if it was not particularly important in the Multinational development but it could easily have been important under different circumstances. For example although communication problems had a relatively minor impact on the schedule this was because communications between the different groups involved in Multinational were generally good.

The events in the commentary are also categorised according to the key underlying influence (see section 5.1.4). In this way the key influences affecting the software development at X are identified. Section 7.4 contains a preliminary analysis of these key influences to assess whether the development in X was more affected by technical or social influences. The twelve key influences are further analysed in chapter 8.

7.1 SOFTWARE CONCEPTION OR INCEPTION?

This section considers the decision to build an internationally distributed database system for the Multinational aspects of X's business. It asks the question: were the foundations for this software development based upon rational decisions reflected in the IT Strategy and Requirements Study or were there other influences?

So, why did X-Group decide to build a system for the Multinational Operations element of the business? What was the scope of the system and how was this established? In particular was the system designed to meet the operational needs of Services and underwriters or management information needs? Why did they decide to adopt a client-

server architecture based on PCs and an internationally distributed database? Why did they decide to adopt a prototyping approach for phase I but a structured approach in phase III? Why did they choose to implement initially in the UK, Holland and the States and how was the division of work into phases arrived at?

7.1.1 Why build a system for Multinational?

The formation of X as a separate company and the arrival of Tim as MD provided the impetus needed for X to develop its own system and the IT Strategy really just reinforced this decision. Market considerations and the inadequacies of existing information systems indicated the need for a new system for Multinational even if this was difficult to justify in cost benefit terms. However the nature of the business, with special requirements for each large client, and the attitude of senior management explains why there had been no previous successful large scale computerisation of this area of X-Group's business.

a IT Strategy (Important - Method)

There was a view that the IT Strategy study was just a formality to get the money. In support of this, the strategy looks very like Colin's existing vision, the points in the strategy about data accuracy were not heeded and when the strategy document became out of date it was not reviewed.

As with many management information systems, it was actually difficult to justify the system on cost benefit grounds. The projected cost of the system was less than the notional cost of buying the service from XUK but the benefits were hard to quantify especially when compared with the cost benefit analysis for the Marine system. In practice, partly because of the Multinational system, there were significant increases in income without increases in staff.

b New MD (Critical - Individual)

Tim, the MD for the new company was a driving force for computerisation. He got approval from X-Group for the expenditure despite financial constraints and problems with demonstrating benefit.

c New business (Critical - Organisation)

The creation of X to focus on underwriting worldwide insurance risks for multinational customers meant there was a need to draw together information from around the world. A system for Multinational was regarded as critical to getting the business going.

The overall information systems for X were poor and prevented X from achieving its market potential or managing the business properly; the quality of overall management information for X actually went down when X was formed. X depended for its IT on local operating companies (ie XUK in the UK) but the nature of X's business was quite different from XUK and the UK mainframe system was inappropriate for Multinational business. New systems would allow X to assume more direct control over its own IT direction.

Where the mainframe systems were inadequate 'home built' PC systems were used. The IT Strategy study expressed concern that these business critical systems had not been subject to 'conventional system disciplines'. The Q&A system was such a system but it was in use for well over three years and was not overly criticised by its users although it was limited by its technical architecture.

d Market considerations (Important - Domain)

In 1991 the international insurance market was well defined with few players and competition was on the basis of quality. This and the difficult state of the market meant that IT was critical in meeting the competition when establishing a new company such as X.

e Nature of business (Important - Domain)

The complexity and one-off nature of the business mitigated against computerisation of Multinational. Multinational business consisted of a small number of long-term complex cases (about 6000 policies worldwide), processed in many countries with service a key factor. As a consequence, information had to be tailored to each client's requirements making automation difficult.

f Attitude of senior management (Critical - Organisation)

Despite a strong case from both market considerations and the incompleteness of existing information systems, the drive to computerise Multinational Business at management level

was not particularly strong. There was an attitude that getting the business was more important than controlling the business and so senior management could not see the need for a computer system. Perhaps it was the one-off nature of X's business, which made it different from the rest of the X-Group, that had allowed this attitude to persist.

g Attitude of underwriters (Important - Individual)

In contrast, Keith and before him another underwriter, felt so strongly that computerisation would enable them to do their jobs better that they developed the first two successful Multinational systems despite the fact that they were neither experienced software developers nor was it within the remit of their job. One of the factors that influenced Keith was his previous experience of computerised offices and the technical management job he had had in Indonesia. Despite an absence of organisational commitment, Keith continued to develop and enhance the Q&A system beyond his immediate requirements. Many of the underwriters used the Q&A system but the management reports produced from it were ignored.

7.1.2 What affected the choice of system boundary?

The original focus for the system as presented in the IT Strategy was for management information for Multinational but the Requirements study extended the system boundary to include much of the functionality of the existing operational system, Q&A.

Unfortunately the operational needs of Services were not considered, perhaps because this group had only just been set up. It was not until the system had been in use for over a year that this deficiency was addressed. The scale of the system varied according to who was driving the vision. Under George's influence, the Multinational system became key to the success of X despite the serious problems with the use of the system highlighted in the Post Implementation Review.

a IT Strategy and Requirements studies (Important - Method)

The IT Strategy report recognised the need for both operational and MIS type systems and suggested that in the short term the Q&A system could continue to be used for data entry. Three systems were identified for immediate work: Financial Consolidation and Reporting,

Multinational and Non-Marine Reinsurance. However the Requirements study does not appear to have been constrained by the recommendations in the Strategy and the resulting requirements were larger and included reinsurance and much of the MIS. The Q&A system was replaced but this does not appear to have been made explicit and not all features of the Q&A system were included in the requirements for the new system.

b Development of Multinational Services Unit (Important - Organisation)

The Services unit for Multinational had only just been established when the Requirements study was undertaken and prior to that the underwriters had done all their own data processing. This was probably why the operational needs of Services were not really addressed in the development of the system. It is unlikely that the different location of Services was a barrier because X-Group were very used to Services operating at physically different locations.

c Implementation of phase I & II (Critical - Technology)

The actual implementation of phases I and II highlighted the limitations of the system for Services and in phase III an exercise was undertaken to decide how best to improve the efficiency of their work.

d Changes in vision (Critical - Individual)

The scale of the Multinational system and its relationship with the mainframe systems of other companies in X-Group varied through the life of the project mainly as a result of who was providing the vision of the system. At one end of the spectrum, the Multinational system was a core system for X replacing existing mainframe systems making X more independent of the X-Group; at the other extreme, the Multinational system was just a management information system drawing data from the mainframe systems.

Initially Colin in the strategy work favoured the replacement of mainframe functionality but when Gordan took over he proposed drawing data from existing mainframe systems.

With George's arrival in phase II the vision for the system expanded; IT was taken much more seriously in X, leading to a long term aim of seriously reducing X's dependence on mainframe systems of the X-Group. So, while under Gordan's influence, the Financial Accounting sub-system was a fairly simple extension for Holland because there was no

business benefit in replacing the accounting systems in the UK and US, a year later, the Financial Accounting sub-system was being designed to be installed in the UK.

e Post Implementation Review (Potentially important - Method)

The Post Implementation Review had a curious impact. Instead of the senior management saying we want less to do with IT and Gordan which is what might rationally have been concluded from the report, they promoted Gordan and decided that they wanted more IT and to become independent of the XUK mainframe systems.

f Organisational change (Important - Organisation)

By mid 1994, X had changed into a more serious established company and with this came an acceptance that IT was key and a recognition that success with the Multinational system could make the company number 2 in the world. IT had become both a driving and supportive force within X as the systems raised people's expectations of what could be achieved.

7.1.3 What affected the choice of systems architecture?

With the benefit of hindsight, a client-server architecture seems entirely appropriate but at the time (1991-92) this technology was not so prevalent. One of the driving forces seems to have been the choice between the PC and the mainframe. The mainframe solution represented a loss of control by X to XUK, while the PC meant a quick system, owned by the business and developed by GIS. However technically inappropriate the PC solution, it would have been attractive to X, particularly to Tim who was trying to establish X as a new business independent from XUK.

The key decision to use a PC rather than Unix-based server for phases I and II was influenced by the increasing power of PCs, the team's familiarity with PCs and the relative costs, as use of a PC server allowed phase I to go in under budget.

The nature of X, operating in a global market out of many locations, favoured a distributed database solution. The change implicit in a new company was one of the reasons for choosing an open systems architecture.

a IT Strategy and Requirements studies (Important - Method)

The original shape of the system using a PC LAN appears in the very first part of the strategy. The decision to use a relational database and fourth generation tool came, at least in part, from Tim but the choice of tool was fully investigated in parallel with the work on the Requirements study. This work involved careful investigations into hardware, communications, database and development tool.

b Individual vision (Critical - Individual)

The PC LAN model originated from Colin during the early work on the strategy study. Prior to the formation of X, XUK was working on a mainframe-based solution for Multinational Business. Tim is reputed to have said 'none of this mainframe nonsense we need a dedicated PC system'. The decision to stop the mainframe development also encouraged GIS to take a different approach and consider using a fourth generation language.

c Image of PC LAN vs mainframe (Critical - Organisation)

On X's part there was a great desire to go for a 'dedicated PC system' possibly as a backlash against the 'nice IBM mainframe' or because they saw it as a way of getting control of their computing. Viewed dispassionately, the decision between a mainframe and a PC should depend upon the power needed for the problem (RAM and mips), but there was a belief that a mainframe would take longer (the earlier requirements study took one year) and would have difficulty coping with the complexities of multinational insurance whereas a PC would be fast, flexible and owned by the business.

d PC capabilities (Important - Technology)

The team chose client server architecture because they believed it would enable them to build something quickly to compete in the market and that use of PC LANs would allow incremental development at reasonable cost. The influence of PC technology is also reflected in the prominence given to the GUI interface in the business objectives part of the IT Strategy study. One of the outcomes of the project was that it introduced the concept of a PC network infrastructure into X offices.

The development team was probably influenced by a move in GIS away from packages back to building systems using the new powerful PC-based development systems. Although a mainframe own build approach was rejected during the IT Strategy because it was too expensive, they adopted an own build client server solution even though there was no evidence that it would be cheaper. The development technology facilitated the use of prototyping, but also made prototyping more necessary because the developers did not know its capabilities.

The Multinational system was developed at a time when the power of PCs was developing fast. The increase in power was such that they did not have to use Unix for the server in phase I as planned but did have problems running graphical applications on the client even though by this stage they were using 486 chips instead of 386 chips as planned.

e Experience of developers (Critical - IT profession)

One of the problems with early attempts at computerisation was that the XUK computing people were used to the XUK way of doing things (high volume, low value cases) which did not match the cases handled by X. GIS (Group Information Services) had wider experiences because they serviced a number of X-Group companies. In terms of architecture, GIS favoured PC based systems because they serviced the smaller companies in X-Group while XUK preferred mainframes because they were used to large operations.

The choice of a PC based LAN rather than Unix was initially a temporary measure while they continued to evaluate the Unix options but they were already familiar with PC technology and at least one of the key programmers did not like Unix.

f Financial considerations (Important - Resources)

Unix was also expensive (almost half the original hardware/software budget for phase I) and if they had used Unix for phase I it would not have come in under budget.

g X needed to operate as a global company (Critical - Domain)

The system architecture had to support the movement of information between offices and countries. A Multinational programme typically involved policies in many countries each reflecting local conditions, local collection of premiums and claims submitted world-wide. All this information had to be consolidated in the issuing country. Operating

internationally also involved mixed currencies and national financial regulations. This was a fairly specialised market so there was not an off-the-shelf package and X would not have wanted to use a competitor's system because they hoped to use IT to get ahead of the competition.

To handle these requirements the system consisted of distributed databases with each database containing information on programmes controlled locally and business serviced locally. Initially information was transmitted electronically overnight but following the installation of the Unix servers they planned to use the Sybase Replication technology. The Multinational system thus had to link to a wide variety of local systems and operate within different IT cultures.

h Structure of X (Important - Organisation)

Not only did X operate out of offices in many different countries, the organisation was dispersed within a country. Despite the inconvenience from an IT perspective, the global and widely dispersed nature of X was one of its strengths. Moreover the organisation was used to working remotely and transmitting information between physical locations and the autonomy of many of the local offices was conducive to a distributed system.

i Allowing for change (Important - Change)

That change was part of life for X and that this would affect IT solutions was recognised in the IT Strategy study. While there was not much evidence that this influenced the way the system was built, it certainly was an issue in the choice of system architecture and was one of the reasons for adopting an "open systems" product such as Sybase. The resulting system seems to have been resilient to change.

7.1.4 What affected the choice of process model?

GIS had little experience of using rapid development approaches and was naturally risk-averse but despite this the Requirements study established the evolutionary prototyping approach that was used in phase I. The success of the Requirements study using a small committed team and the personal relationship between Gordan and David contributed to

this decision. The Requirements study was also responsible for the ambitious decision to install in three countries in phase I based on business demands not feasibility.

Full evolutionary prototyping was not proposed for the Financial Accounting sub-system because of its complexity but the appointment of Colin as project leader led to a very traditional approach to development. The appointment of a methods expert as Multinational project manager and the organisational reaction from the technical problems of phase II, resulted in them adopting a more traditional approach to the whole of phase III.

a GIS development approaches (Potentially important - Method)

GIS had a development method, which included traditional, structured and rapid development approaches, but this was regarded as outmoded in some areas and projects were expected to develop their own procedures. Multinational was regarded as a modern forward-looking project and the team chose to use rapid development techniques which had not been used by GIS for a major external project. This decision seems to be inconsistent with GIS's reported fear of failure as prototyping was quite new for its time and would not have been regarded as risk free.

b IT Strategy and Requirements studies (Critical - Method)

The IT Strategy set the methodology, an aggressive timescale and costs but these were based on producing a different kind of system 'using existing and off the shelf IT components'. It was the Requirements study which established that evolutionary prototyping would be used for the development. The study also divided development into two phases and decided that UK, Holland and USA would be included in the first phase. The size of the system seems to have been driven by a desire to 'get as many things that seemed important into the first stage' and data dependencies were not realistically considered.

The IT Strategy also correctly predicted the importance of data accuracy and the need for the right operational procedures and warned that 'significant management effort and commitment' would be needed to address these issues.

c Location of business (Critical - Organisation)

The decision to include UK, Holland and the US in the first phase was based on the fact that these were key countries in X's business and did not take account of the huge cultural differences in the way the business was conducted in America compared with the UK and Holland.

d Small and fast (Critical - Belief)

The Requirements study was completed in a remarkably short time. Key to this was a small team of committed and enthusiastic individuals. In choosing the development approach they would have wanted to match this success.

e Business involvement (Critical - Individual)

One of the key factors in planning the Multinational project was that it should be managed as an equal partnership between the business and IT people. This allowed the business to have ownership of the project but avoided the problems there can be in handling the technical aspects of a project which is managed solely by the business. A senior person, David, was seconded full-time from the business side who had the additional advantage of having been involved in the previously failed attempt at computerisation by XUK. Senior people were also involved on a part-time basis from Holland and the USA. One reason for seconding someone as senior as David was that it demonstrated organisational commitment. One of the key reasons for continuing the partnership during the development of phase I was the very good, equal personal relationship between Gordan and David.

f Project scope (Important - Domain)

The complexity of the Financial Accounting sub-system was why it was initially decided to use a more traditional development methodology with throwaway prototyping. There were some difficulties in managing this different approach when it interfaced intimately with the rest of the Multinational system, which was still being developed using a more flexible approach. It was also difficult to manage user expectations.

g Appointment of project managers (Critical - Individual)

When Colin was appointed as manager of the Financial Accounting sub-system of Multinational he seemed to be determined to take a very traditional approach to project

management introducing all the conventions of a traditional life cycle (such as signing off specifications, programming to a module spec). Although the Financial Accounting sub-system was running behind schedule, I feel this change had more to do with personalities than just the nature of the problem.

Although Jane was not recruited because of her strong methods background it was very influential in her subsequent management of the project. It was a key factor in the decision to introduce a more structured approach to both method and team organisation in phase III.

The change in process model seemed more organised and formal but Gordan summed up the impact as 'It's all traditional, sequential, boring.'. Unfortunately I did not continue to follow the project for long enough to observe the impact the change in methodology had in practice.

h Organisational experience of development (Important - Organisation)

The development method in phase II seems to have been less well defined. I think they just continued to develop in the same way but without the prototyping workshops to give the process structure and a timetable. A contributing factor to the problems with phase II was the change in emphasis in development strategy that was not recognised clearly, or early, enough by the development team.

The Post Implementation Review of phase I recommended a more throwaway approach to prototyping during phase II. Gordan was still enthusiastic about the evolutionary approach but the analysis for the extensions to phase II was undertaken as a series of mini studies with limited user consultation. By phase III a more traditional development approach was being used, not necessarily because it was more appropriate, but because their fingers had been burnt in phase II and so they could not take chances.

7.2 SOFTWARE CONSTRUCTION OR EVOLUTION?

This section focuses on how the three phases of the Multinational system were built. It looks at what was actually built, how it was constructed and the development process and asks: was this a managed process based upon rational decisions or did the process evolve shaped as much by social factors as technical requirements?

So, what factors influenced the content and success of each phase? What affected the design and reliability of each phase? What influenced the process and management of each phase including the transition between phases? What influenced the composition of the development group?

7.2.1 What affected the functionality that was delivered?

In phases I and II the system was primarily a management information system and did not satisfy the requirements of operational users (either services or underwriting). The key reason for this emphasis was the prototyping approach and the people involved. In phase I the system designed was much bigger than intended primarily because of the lack of experience of prototyping and the enthusiasm of the individuals involved. The functionality delivered was influenced by the technology both in adding functionality and limiting what could be implemented. There were some detailed requirements that were difficult to define because of the nature of the problem domain.

There were largely unfounded suggestions that the missing functionality required by Services was not provided because of time pressures or because Services was structurally remote. However, data quality problems probably did limit the reporting capabilities provided early in the system and after phase II the instability of the system made the users wary of adding functionality.

a Prototyping methodology (Critical - Method)

The prototyping methodology had some very positive effects on the development. A lot was achieved in a short period of time and the project came in under budget. Prototyping was a catalyst for change within the organisation and encouraged discussion of business issues. It also allowed people to have a say in the development and so produced a system which was more oriented to the requirements of some users. As with most prototyping projects the impact of having a working system to comment on was much appreciated although expectations of the time taken to produce an operational system had to be managed. There was greater commitment to this system than previous attempts at computerisation.

The team had difficulty setting a reasonable boundary for phase I. This was partly because the initial scope was larger than they realised and also because they had difficulty stopping the prototyping process. Given the existence of the Q&A system it is surprising that they were not more aware of the size of the scope but I think this reflected a lack of clarity over the aims of the system. Shortly after phase I went live Gordon felt that the outcome justified the ambitious nature of phase I, because the linkages between countries made the system very impressive. One of the typical prototyping problems, which all those involved reported, was that the business users did not know when to stop and continued tweaking the screens.

Despite the increased functionality, phase I did not satisfy Services because it did not include all the functionality of the Q&A system. This was either because the system was not really intended for Services or GIS did not ask the right questions. They started with a blank sheet of paper letting the data dictate the approach, rather than studying existing procedures (this was actually undertaken as part of phase III). I do not think the involvement of business people removes the responsibility from the developers to see the big picture and point out to the business people the consequences of their decisions. The fact that the users did not seriously use the prototypes probably meant that these shortcomings were not picked up early enough for enhancements to be included in phase I.

In the development of phase II there were not the same prototyping workshops and, probably as a consequence, changes were made in phase II that the users were not aware of and did not like.

b Business user involvement (Critical - Individual)

A wide variety of people were invited to the prototyping workshops but attendance varied and in the initial workshops there was a greater emphasis on getting management rather than underwriting input. David's view was that technical managers should know what they wanted of the system and the underwriters' role in design was to make sure the system was usable. I think it was this decision that made both phase I and II more management tools than operational systems. Unfortunately the managers overlooked the effort of putting data in in order to get information out. However, given the lack of interest in the development shown by the underwriters, I am not sure that their participation in the workshops would have made that much difference.

There were problems with the workshops: attendees did not always read the paperwork, had not necessarily attended previous workshops and had different interests and objectives. The workshops came to conclusions because of David's leadership but the system was probably effectively designed by those working full time on the project team.

David was senior but remote from the underwriters. He was also very 'strong-willed'. Keith was not representative of underwriter interests because of his commitment to computerisation and was considered by some to be out of date. While Lucy might have been expected to push the Services perspective she had no previous experience in multinational insurance and was very much David's junior. In contrast when Jenny replaced David, her Services background meant Services started being heard.

Another consequence of David's involvement was that phase I was bigger than intended because he stressed the user benefits and pushed to get as much into phase I as possible. However his seniority, authority to take decisions, and good relationship with Gordan did mean that phase I achieved a lot in a short timescale.

c Development team enthusiasm (Critical - Belief)

The commitment and enthusiasm of the development team also contributed to the development of a much bigger system than had been planned.

d Technology (Important - Technology)

One of the consequences of involving business users in design in phase I was that technical issues were considered after functionality. This was regarded as an advantage because too early consideration of technical constraints would 'stifle the imagination'. However, the technology was an important factor in the growth of the functionality of the system and was one of the reasons why it became the core system for X. The McDonalds system and the Replication capabilities of the Unix version of Sybase were good examples of how technology enabled new functionality. In contrast, the continued problems with ad hoc reporting and the difficulty of using Q+E effectively illustrate the impact technology could have on functionality.

e Nature of business (Important - Domain)

There were some areas where functionality was difficult to define. The reinsurers would not define their location requirements. Larger clients had to be handled individually in deciding exactly the level of detail to record in the system. For McDonalds, their biggest client, they actually produced a complete subsystem based on a cut down version of Multinational to handle McDonalds' specialised information requirements.

f Time pressures (Potentially important - Resources)

Lucy reckoned that shortage of time and the pressure to go live was one reason why functionality for Services was cut in phase I. Although some requirements were dropped because of time constraints, I did not get the impression from anyone else that time pressures were the reason why phases I and II were not more useful to Services.

g Structure of X (Potentially important - Organisation)

Following the normal division of work in X-Group, Multinational Underwriting Services was created in Essex separating the operational side from the underwriters at board level. Moreover the head of department for Multinational Services did not seem to be involved in their work (perhaps because it was so different from the other servicing units at Essex).

Given this structural separation the emphasis on the underwriting side in phase I is easy to understand but I do not think this was an important influence.

h Data quality (Important - Method)

Data quality problems were probably a contributing factor in the lack of impetus on the development of reporting capabilities because many of the reports would be useless without good quality data. The impact of data quality on development was illustrated by the success of the McDonalds system, which did not involve existing data because it was a new programme.

i Experience with phase I & II (Important - Organisation)

One of the consequences of the problems with the system was that all users wanted stability, not more automation. I do not think this was because of the recommendation of the Post Implementation Report to get benefit out of the system before further automation, I think they just wanted a working system.

7.2.2 What affected the internal design?

In phase I the use of prototyping led to a focus on external design so internal design was neglected. However, it is not clear that the more traditional approach adopted by phase III would necessarily change the focus. One problem was that the technology they used made development easy and did not provide support for good internal design. Time limitations meant that the design flaws in phase I could not be corrected until phase II. A shortage of time was also blamed for the absence of full technical documentation but prototyping was probably a contributing factor.

The quality of design throughout the Multinational system was greatly influenced by the ability of the individual developers although experience was also important. The abilities required of development staff were many and did not always coincide with what they liked doing. Leaders of the developers should probably have been more aware of the design but experience of phase III shows that this was not easy.

a Methodology (Important - Method)

One of the most serious consequences of the prototyping in phase I was the poor database design. The screens were designed almost as if it was a throwaway prototype without carrying through the entity model developed in the requirements study into a detailed database design. The database grew in an ad hoc way so that its structure was not relationally sound. These problems were not caused by an initial lack of design but I think arose through a lack of experience with prototyping and fourth generation development tools. An outline design was produced before starting the prototyping, which included an entity model, content of screens and a summary of processing. However during prototyping the focus was on the screen design and associated processing and the system was designed by directly coding in Powerbuilder.

There were also problems with the software design although these were less acknowledged. The system grew rather than having a modular design and did not handle validation and errors in a standard way although visually it was generally consistent. Within phase I the design was not revisited until it was too late to do anything about it and from an external perspective the system was fine if slow. The exception was reinsurance which was rewritten during phase I because there were difficulties in defining the requirements.

The quality assurance review of phase I highlighted non-adherence to standards as one of the causes of the deficiencies in this phase. However there were no indications that non-adherence to standards was the cause of the database design problems. There were differences in internal development style particularly in phase I, but these were due to the newness of the tool, making overall internal standards difficult to develop.

The newness of the tool was also one of the reasons why technical documentation in phase I was deferred (with Quality Assurance agreement). Unfortunately this seems to have resulted in a lack of emphasis on technical documentation for which the team continued to be criticised. Although resource constraints were officially blamed, I suspect it was because a design document was not found to be necessary for prototyping. When I left they had just started to re-engineer a design document for the system for use during maintenance.

Unfortunately I did not observe long enough to discover how useful this would be, but up to

the point of my departure the developers had not highlighted their need for such a document. One of the advantages of fourth generation tools is that they can be self-documenting but there was no evidence that the team were using such features - I was never shown documentation produced from the system.

Although the general consensus was that the prototyping methodology was the cause of the design problems, it is interesting that the Financial Accounting sub-system specification did not appear to have an entity model. This was despite Colin taking over with his more traditional approach to development, his greater emphasis on formal documentation and the data modelling problems in phase I of the Multinational system. For this reason I have classified the impact of methodology on design as important but not critical.

b Technology (Important - Technology)

One of the reasons for the design problems of phase I was the 'mirage' of tools such as Powerbuilder, which can make system building seem easy. The use of an object-based design in phase II was not always possible because they were learning as they went and the technology did not directly support the approach. Adherence to design principles depended upon the influence of the lead developers and the reporting side was separately designed by other developers. They did not use an object oriented language such as C++ because of lack of expertise and PowerBuilder was easier to learn.

c Time pressures (Important - Resources)

Pressures to meet the schedule in phase I had serious implications on the internal design of the system; when they realised there was a problem with the database design there was no time to rewrite it. These same pressures also meant there was not time to explain or enforce standards leading to some differences in internal design.

d Development staff (Critical - Individual)

The experience and abilities of each of the development staff had a considerable impact on the design of the system. In phase I they got to grips with the new tools quickly, but I suspect their inexperience with the tools, coupled perhaps with some inexperience in design, contributed to the poor design of phase I despite the high abilities of some staff. Phase II design benefited because the staff had greater experience with the tools. In some

cases, ability was more important than experience in determining success; the Unix migration was completely successful despite the inexperience of the developer but reinsurance had to be rewritten partly because the person assigned to it was 'out of his depth'.

e Expectations of development staff (Critical - IT profession)

A lot was expected of the development staff: good technically (you could trust them to find and implement a solution), good analysts (could see things from the user perspective), have a wide view, good communicators both externally and within the team, committed. The numerous minor problems with phase II also indicates that they were not 'completer/finishers'.

The comment that the programmers liked variety not just 'boring old development' is revealing. Redesigning the system would have been seen as interesting and challenging but completing all the little details would not. The programmers talked about how a flair for programming should not stop you doing other things and how they preferred working with people rather than 'grappling with code on my own'.

f Leadership of IT professionals (Important - IT profession)

One of the consequences of prototyping is a non-hierarchical structure because each of the developers needs to be able to take a lot of decisions and this was reflected in Gordan's approach. The lack of internal design in phase I was a reflection of Gordan's trust in the competency of the team which given their lack of formal training was perhaps misplaced.

The rewrite in phase II showed how easy it is for the leader of IT professionals to become unaware of exactly what they are doing. At X, there were ongoing debates over how much a project leader should be involved in the detail of design. I think that the lack of an entity model in the Financial Accounting sub-system design shows how hard it is for those in senior positions to really influence technical development even if they want to.

7.2.3 What affected the reliability of the system?

The reliability of the Multinational system depended upon who you asked and when. The development team were very positive about phase I but retrospectively users were very critical. Phase II was regarded by everyone as unreliable but some users were far more critical than others and the development team felt that its good points were not recognised.

In phases I and II there was a lack of time and commitment to testing on the part of both users and developers. This was more significant in phase II because prototyping meant the system was exercised during development in phase I. By the time of the Unix conversion, they had all learnt from the experience of phase II of the importance of testing. The quality assurance function within GIS does not appear to have contributed either positively or negatively to the reliability of the system.

There were major problems with Sybase in both phases I and II which had a significant impact on reliability and which were largely out of the developers' control.

The Post Implementation Report highlighted the absence of effective project leadership as the biggest problem with phase II. Although Gordan's time was stretched and the team leader he appointed had difficulties establishing his authority over the team, the situation was not clear cut and this provided an effective scapegoat hiding some of the other problems. However, individual developers were critical in either alleviating or aggravating the problems of phase II. The complexity of the problem was important.

a Impression of reliability (Important - Belief)

The reliability of both phases I and II depended upon who I asked and when I asked them.

The view from the development team after phase I had gone live was that it was a good system and this view was echoed by the Post Implementation Review. In contrast the users were all (underwriters and Services) very negative about phase I retrospectively. They complained about the system having errors and miscalculating data and about it hanging and crashing. Although some of these problems were due to Sybase and were also reported by the developers, they did not appear to view them as severely.

With phase II there was agreement that it suffered from a large number of problems when it went live. Many of the problems were relatively minor but the situation was aggravated by the number of bugs and because bugs re-emerged that had already been fixed in phase I.

Different groups of users reacted differently to the problems of phase II. Although the underwriters used the system less than Services and were less dependent on it in their everyday life, they tended to be much more vociferous about the errors. This reaction may have been because of the different nature of their use but presumably the Holland underwriters had a similar use and they were more relaxed about the problems.

The situation in the City was exacerbated by Jenny who saw herself as the customer despite being committed to the system. In contrast Keith was always very positive. There was also some feeling that system problems were blown up and used as an excuse by senior management in Essex. Despite the problems, the good relationship between GIS and X was largely maintained.

b IT profession's definitions of success (Important - IT profession)

The developers view of phase II was that it was a technically better system and that they were nearly successful with the installation. They were frustrated that the users failed to recognise the technical quality of the system because all they could see were the errors.

c Methodology (Important - Method)

Testing is always a problem with prototyping because although the system automatically gets tested as part of the prototyping process, it is also necessary to do systematic testing at both unit and system level after prototyping is complete. There are indications that phase I suffered from this problem but that the system was fairly thoroughly tested during prototyping. This testing was largely done within the team (by Keith and David) and so did not really ensure user acceptance. The operational users found it difficult to use because there was never a fully operational test system.

In phase II business users were not part of the development team in the same way so that there was virtually no testing by users. Acceptance testing was not built into the plan because of the belief that there were no really new requirements. Phase II was tested by

the development team but they found it difficult to test thoroughly because it was a Windows development and as in phase I it was difficult to distinguish between unit and system testing.

One of the other main differences between the installation of phase I and phase II was that in phase II there was an operational system and the team had the added complication of working on two systems. Configuration management is often a problem in such cases but does not seem to have been particularly problematical.

With the Unix migration they had learnt their lesson and testing was exceedingly thorough; they dedicated resources to testing and included parallel running and stress testing. The migration went fairly smoothly but took a lot longer than it needed to.

d Resources (Important - Resources)

Phase I was not subject to rigorous user acceptance testing despite having the manager of Multinational as part of the team. The underwriters and Services did not really exercise the system properly because management were not committed to testing, the departments were busy and short of people.

e Quality Assurance (Potentially important - Method)

GIS's traditional internal quality assurance function appeared to operate largely in a monitoring role, checking that projects had standards and adhered to them. Quality Assurance was such that it was possible for the project team to cut corners and there is evidence that Quality Assurance was not generally regarded as critical by GIS. The most visible aspect of Quality Assurance was retrospective post implementation reviews but these were not particularly effective in bringing about change. However the problems of phase II were not attributed to failures in Quality Assurance.

f Sybase (Critical - Organisation)

There were two major problems with Sybase on Novell that eventually led to the migration of the larger servers to Unix. The first problem occurred during phase I and eventually Sybase solved this by installing a different communications protocol but not before the team had spent two weeks installing a trap in the software to report the error rather more gracefully to the users. The second problem occurred intermittently during phase II. It

could not be reproduced and appeared to go away for weeks at a time only to recur several times on one day. In the end the second problem was only solved by migrating to Unix at great expense.

These two problems illustrate the difficulties of working with a complex layer of software. There was nothing that the development team could do to resolve the errors except hope that Sybase could solve the problem. As X-Group is a large company they did have some influence with Sybase although they did wonder whether they would have had better support from a more UK-based company such as Oracle. One of the frustrations for the developers was that the users could not usually differentiate between application and system software problems.

g Technology (Potentially important - Technology)

Apart from the problem with Sybase there were few problems with the technology and the networking worked well despite international differences in ISDN implementations.

h Project leadership (Important - IT profession)

The Post Implementation Report highlighted the absence of effective project leadership as the biggest problem with phase II. Although this was undoubtedly a problem the situation was not clear cut and it provided an effective scapegoat hiding some of the other problems.

David and Gordan's number two, Scott, left the project as phase II got underway and Gordan and his senior analyst, Stuart, were involved in new developments in addition to their work on Multinational. Despite this Gordan was still managing the Multinational project. Although a strong project leader coming in at this stage might have averted the installation disaster, in practice the problems occurred earlier in the development and stemmed from a lack of control over what was being built. In addition there was a lack of recognition by the team of the problems they were facing, perhaps stemming from a feeling that they were supposed to be getting on with things, rather than Gordan being unapproachable.

i Development staff (Critical - Individual)

Individuals in the development team played a big part in the outcome of phase II. When Stuart was appointed to lead phase II, the team was operating on a basis of trust with a

non-hierarchical structure. Given Stuart's character it would have been difficult for him to change the style at this stage. He had doubts about the team's ability to make the deadline but was prevented from making a realistic plan because of his lack of technical knowledge and lack of authority in the team. He felt that given the environment it was the responsibility of the individuals concerned to realise the complexities and raise doubts themselves with Gordan.

Two of the developers in particular helped the development to emerge successfully from the fallout of phase II. Gordan managed to absorb the hassle of the bugs protecting the team without going on the defensive and won the continued support of X management. Alan, despite being a junior in the team with little formal training in computing, managed to develop an effective system for handling bugs and getting them fixed even if the users sometimes found it inefficient reporting errors through Keith.

In contrast Kevin's lack of knowledge meant that errors reported to him bypassed the system whatever their severity. However this escalation may have also been due to his seniority because the same thing happened when Tim was involved in problems.

The individual developer's understanding of the business was generally regarded as very important to the development.

j Nature of the problem (Important - Domain)

It is interesting to compare the implementation of the Unix migration with phase II. On the face of it, the Unix migration had the potential to be as problematic as phase II and the team were making the same reassuring noises before hand. In practice the Unix migration went very smoothly and with very few problems. The developers were very careful with the Unix migration because they knew that they could not afford another disaster, however my impression was that unlike phase II it just turned out to be a much easier job.

7.2.4 What affected the schedules?

There were scheduling problems caused by the lack of experience and difficulty of managing a prototyping project, but prototyping also enabled them to deliver a much larger

system more or less on time. The first phase of Multinational came in on schedule mainly because the senior business user had experience of a previous project drifting. However the tight schedule of phase I did put a lot of pressure on staff leading to a loss of momentum between phases I and II. The loss of momentum was also fuelled by a loss of vision for the development.

The availability of staff impacted the schedules throughout the development particularly the time taken to recruit new project leaders and the involvement of the project manager in other projects.

The team consistently under-estimated the size of the problem and although this was partly because of a lack of familiarity with the technology, it was also because they under-estimated the complexity of the task. The different software systems used sometimes helped them meet tight deadlines and sometimes proved to be more difficult to use than anticipated. The schedules were also affected, but to a lesser extent, by organisational change and breakdowns in human communications.

a Prototyping & project management (Critical - Method)

The fact that phase I delivered on time despite being significantly larger than planned was attributed to the success of the prototyping methodology. The only reason the team could claim to deliver on time was because the original deadline was a bit vague and the system was delivered in two parts at the end of May and June.

Project planning did take their inexperience with the tools and prototyping into account. However, although progress monitoring at various levels was in place, they did not use it to establish a cut-off point to changes and let the project run on for too long so it was significantly larger than planned. There was a lot of pressure to deliver something before David left the department and to achieve this they worked long hours and phase I was installed although they knew that it needed improving technically.

Immediately following the installation of phase I there was a loss of momentum partly because X could not decide on priorities, but the developers needed a breathing space because they had been working at an unsustainable level to meet the phase I deadline.

Phase II began without detailed plans although there was an overall plan. Although the Post Implementation Review said this plan did not include 'sufficient contingency for the novelty of the tools and the team's lack of experience', I think it is more likely that it suffered from second system syndrome in that they thought they knew what they were doing. For example they under-estimated the difficulty of converting from phase I to phase II. It also sounds as though progress monitoring was less rigorous than in phase I and was based largely on trust, and, during phase II, some of the team were still working on enhancements to phase I which were made live during phase II. Phase II was launched before it was ready mainly because there was pressure to deliver on time and the delivery date had already been put back, but also the developers genuinely thought that it was alright, perhaps another example of second system syndrome.

b Business user involvement (Critical - Individual)

Phase I was fortunate in that David had already experienced a drifting computerisation project and so knew the importance of keeping a tight control on progress.

c Loss of vision (Important - Vision)

One of the reasons for the loss of momentum between phase I and phase II was that Tim lost his vision of what they were trying to achieve and so could not decide priorities.

d Availability of key resources (Critical - Resources)

Throughout the development the availability of key resources (most often staff) impacted on the schedules. One of the reasons phase I was installed in two parts was because of the departure of a key analyst. The phase II installation went ahead as planned partly because Gordan was in America at the time and so could sort out any problems there and also because delaying the installation would have added at least a month to the delivery date because of Easter.

In June 1994, following phase II, the X Directors decided they needed to appoint a new project manager, but it was not until later in the year that Jane and Colin could be appointed to the development side and the new year before Neville was appointed to provide clout and focus on the business side. These delays in appointment caused the project to drift through lack of leadership.

Financial considerations also affected decisions: the reengineering for phase II was accepted because phase I came in under budget but when there were all the problems with phase II, X threatened not to agree future budgets.

e Senior management (Critical - Organisation)

Gordan was distracted from leadership of the Multinational project in phase II because he was assigned the Marine investigation by senior management when Marine became part of X and business priorities changed. There was no logical reason why Marine should have been given to Gordan and David cautioned against it. Maybe this decision reflected a perception amongst senior management that Multinational was more or less finished and Gordan had built what was regarded as a successful project for X already, or it might just have reflected the risk taking attitude of senior management. Whatever the reason it looked as though they were set to repeat the mistake with Gordan's involvement with the City project during the period immediately following phase II.

f Underestimating the problem (Critical - IT profession)

Throughout the Multinational project, the team consistently under-estimated the size of the problem. In part this was because they wanted to be responsive to the users. In phase I they under-estimated the amount of work involved in installing the software because this also included installing seven local area networks in the UK, Holland and USA. With phase II they under-estimated the task because it included little new functionality. They may also have over-estimated the ease of using the tools. This was certainly one of the problems with the installation of Q+E for ad hoc reporting. In the case of the Financial Accounting sub-system I think the main reason was that they under-estimated the complexity of the task. This was aggravated by the aims of the system changing as X began to take greater control of its IT.

g Technology (Important - Technology)

The impact of technology on the development schedules was mixed. Although the use of many different IT systems by the organisations which supported X undoubtedly made phase I more difficult to develop, these difficulties appear to have been managed within the schedule. Likewise the technical problems with Powerbuilder appear to have been contained and there was a general agreement that the speed of development was a result of

using Powerbuilder. However, delays in the implementation of both Q+E and the Financial Accounting sub-system were partially due to difficulties of providing desired functionality using package software.

h Internal communications (Potentially important - Communication)

Communication deficiencies within the organisation do not seem to have generally affected schedules. Two problems were reported in the Unix migration: the start was delayed because Gordan did not realise he had authority to proceed and implementation was delayed because the right comms card had not been ordered.

i Organisational change (Potentially important - Change)

Although organisational change can impact on schedules, the only incident I recorded was the physical changes in structure in London in April 1993 which impacted on the networking which had already been installed.

7.2.5 What affected the composition of the development group?

The most important characteristic affecting team composition was ability and to a lesser extent technical experience. Although the people I was interviewing knew of my research interests, team issues were rarely raised. Although capabilities to manage a team were discussed, they seemed to be less important in practice than being tough enough. Seniority rarely seemed to be an issue in team working and so the prototyping methodology helped the careers of people on the team by giving them direct exposure to senior management.

The staffing recommendations of the Post Implementation Review following phase I were not heeded. By the end of phase II, the organisational importance of the system had grown and the resourcing recommendations of the Post Implementation Review were addressed. There were resourcing problems with phases I and II but I think these stemmed from organisational rather than recruitment difficulties.

a Ability & experience of individuals (Critical - Individual)

Ability, and to a lesser extent experience, were very important factors in determining who did what. In building the team, Gordan seemed to look for a blend of expertise covering

technical and business aspects rather than considering personal team-working qualities. He also took into account an individual's commitment to the job.

Keith's situation illustrates how the value put on abilities depended upon who was judging them. During phases I and II Keith was an important and influential member of the team but with Jane's arrival he was sidelined and his attention to detail was not valued. The situation was reversed somewhat with Colin's involvement.

Kevin's appointment shows how important experience could be. His lack of IT experience meant he could not take effective responsibility for some complex managerial decisions and distracted Gordon's attention away from phase II.

b Personality & attitude of individuals (Important - Individual)

Team issues rarely came up in conversations about the development although individuals were often discussed. There was friction within the team during phase II but this was hardly mentioned. There was no evidence that this friction had any particular impact on the problems of phase II. There were problems between George and Terry which were scarcely directly mentioned although I think this was the main reason for the reduction in effectiveness of the IT Steering Committee during phase II.

They talked about project managers needing to be able to mould and motivate a team and have the project management skills of thinking across the project, planning, monitoring and communicating. However in recruiting there seemed to be an emphasis on toughness, which perhaps caused them to ignore other qualities. For example, Jane's technical background had a huge influence on the adoption of structured methods but I am not sure this is what they recruited her for. A key issue in Jane's recruitment was her 'toughness' but there is some indication that this meant she was better at communicating upwards than downwards or sideways. Similarly Neville seems to have been respected for his somewhat bullish attitude to management and this emphasis may have stemmed from Tim's personality.

c Seniority of individuals (Potentially important - Individual)

Generally seniority was not an important issue in the appointment of individuals. Only Jane seemed to be influenced by people's seniority.

d Prototyping methodology (Important - Method)

One of the positive spinoffs of the prototyping methodology was that fairly junior staff on both the user and development side had exposure to senior management; important both for recognition and career enhancement.

e Post Implementation Reviews (Important - Method)

The Quality Assurance review did highlight the lack of leadership on the Multinational team in the Autumn of 1993 but this was not addressed until highlighted again in the Post Implementation Review almost a year later. Possibly one reason for this was that the QA person was not respected by Gordan because he was 'steeped in traditional development methods'.

The recommendations of the phase II review were heeded. It was used by X to bring the resourcing problems to the attention of the Directors who eventually addressed the report's recommendations with the appointment of Jane and Colin in project manager roles, the promotion of Gordan giving him budget authority, the establishment of a steering committee and the secondment of Neville. The report also contributed to the relocation of part of the development team in the City.

f Nature of the Multinational project (Important - Organisation)

Towards the end of phase II, the complexity and size of the Multinational project increased as its role in X became more important. The recommendation to appoint a project manager was a recognition that Multinational had become a difficult project to manage. The size and importance of the project to X continued to grow, leading to greater emphasis on its management.

g Resource considerations (Important - Resources)

GIS intended that Manny should replace Scott in phase II but GIS could not release him from his previous job until January. By the time he arrived Marine was underway and Gordan decided that Manny was needed for that project. Gordan did not really push the

issue hard because the structure made it difficult and he thought they could 'get away with it'. This resourcing problem was picked up as being one of the key findings of the Post Implementation report although the report itself says that 'the scale of the difficulties should not be overstated'. I feel that this was an easy target for blame and diverted attention from their problems of data entry.

The problems of Gordan serving two masters and the impact this had on resourcing were eventually resolved by Gordan being promoted and given the IT budget. The effect of this was noticeable by February 1995 with lots of contractors and consultancy being used in the project.

h Recruitment of IT staff (Potentially important - IT profession)

The project was affected by difficulties in recruitment and retention of appropriate IT staff. When Kevin left they realised that having a business manager with little IT experience as an IT manager had not been successful but could not recruit an appropriate replacement so they split his job. To recruit Jane they had to use a management consultancy agency and candidates were widely different. The person they recruited was very expensive (10% of the total annual spend). Even recruiting junior staff was not easy because it could take up to a year to train up a new member of staff and GIS recognised that staff needed to be moved fairly regularly to retain them. Having listed the problems, I do not feel that the project was actually greatly influenced by the difficulties of recruitment.

7.3 SOFTWARE DELIVERY OR USE?

This section investigates why there were so many problems getting many of the users to effectively use the system. Were these problems because of the way the system was installed or were there other causes? Throughout the study data quality within the system was poor. Why was this?

7.3.1 What affected the use of the system?

The attitudes of underwriters to using computers were widely regarded as one of the key reasons why the system was not used. In the UK, it did not seem to be an attitude to computers that was the problem but some underwriters did not see servicing as their business.

There was no lack of organisational commitment to the Multinational system but some senior individuals were not committed despite the use of prototyping. It is also true that system problems, of phase II particularly, caused people to lose faith in the system.

However, the underlying reason for people not using the system was that it was not useful to them. In the case of the underwriters, the benefits were small and their lack of use meant that data quality was poor and this reduced the system's usefulness to other users.

Usage was also affected by the bad timing of the installation of phase I, insufficient consideration given to the resourcing of data loading and people's normal resistance to change. Training would have helped alleviate these problems but was very limited in phases I and II because there was a belief that prototyping reduced the need for training or a system manual. However, use of the system was encouraged by internal communications, which were generally adequate or good.

a User attitudes (Critical - Organisation)

The attitude of the underwriters was widely cited as one of the reasons why the system was not effectively used by them. The difficulty of getting underwriters to use the system was raised in the IT Strategy in 1991 and the Post Implementation review highlighted that the problem had not really been addressed.

In the UK there is evidence that at least some underwriters were perfectly prepared to use computers. It was underwriters that wrote the systems used prior to the Multinational system and the underwriters adopted other aspects of Windows, such as email, introduced at the same time. Even Donald (who was regarded as one of the less enthusiastic underwriters concerning computerisation) said 'It is vogue to just moan about the computer and all the rest of it, but it's the easy thing to do, but most underwriters, I'd say, when it

comes down to it will say that you've got to have one ...'. There was also a view that the various attempts at computerisation had left a poor image of IT in X but there is little evidence amongst the underwriters in the UK of this.

A more significant factor was the attitude of some underwriters to servicing the cases. Some of the underwriters in the UK believed in controlling their cases while others believed that getting the business was more important. This attitude may have been exacerbated by the decision to create Multinational Services; there was certainly a bit of a 'them and us' relationship between underwriters and Services. However, I think that the difference in attitude to the system exhibited by underwriters and Services was mostly due to the difference in the focus of their jobs.

b Commitment (Critical - Individual)

Throughout the development of the Multinational system there does not seem to have been any lack of organisational commitment to the development. Nevertheless the importance of individual commitment in ensuring that the system was used is illustrated by the problems of America and Claims. Although a senior manager in America was involved in prototyping, and senior management on both sides of the Atlantic agreed on a plan for loading data, the data in America was not effectively loaded and the system had to be relaunched in 1995. Claims in the UK also seemed to suffer from no championship by an individual. There were still problems with Claims in the UK in 1995 although claims were being successfully downloaded in Holland because of the determination of an individual underwriter.

There was a view that one of the problems of phase II was that David was replaced by Jenny who was junior and less experienced. I am not convinced this was significant, both Jenny and Lucy from Services seemed to demonstrate commitment and enthusiasm for getting the system working.

c Prototyping methodology (Important - Method)

The prototyping methodology encouraged ownership by those that were heavily involved and explains the extraordinary loyalty to the system expressed by both Jenny and Lucy.

Unfortunately this ownership did not extend to those who were less involved in the development such as the USA.

d System problems (Critical - Belief)

The errors in the system meant that people, who were not confident with using it or were not convinced it was useful, found it easy to blame the system when they had problems using it even though there may have been human input problems. The number of bugs found in the early stages of version II were used by Jenny to alert X senior management to the system problems and probably made the reaction worse with some managers using the system problems as a scapegoat.

The launch of phase II was particularly catastrophic. It raised unrealistic expectations, reduced motivation and gave the underwriters a reason to be critical. Attitude to the system was contagious - underwriters saw others having problems and so were not minded to put using the system high on their list of priorities. In contrast, Services and Holland, who were more committed to using the system, were much more sanguine about the errors.

One of the advantages of the Post Implementation Review was that it seemed to focus the venom and vitriol but it also seemed to reawaken discontent so that even when most of the bugs had been fixed the users were still fed up with the system and management re-considered its long-term status.

e Functionality and usability (Critical - Design)

One of the reasons why I believe the underwriters did not use phase I or II was that it was of limited usefulness to them. There is a marked contrast with the other facilities provided with Windows, which the underwriters did adopt.

The system was not useful as an information resource because much of the data they wanted to access was still held in manual files or was easier to access manually. In particular historical data about claims was not loaded. It was regarded as unlikely that the system would ever replace paper records. Very few reports were written for underwriters in phase I or II of the system and the absence of ad hoc reporting meant they could not write their own reports as they had with Q&A. Cases were easier to renew once

they had been correctly entered into the system. However, loading the data was not easy because it was time-consuming to enter single bits of information about a programme. In phase I this was exacerbated by the speed of the system but phase II was still regarded as too slow by users.

The system was also limited in its usefulness to Services. The first installation of phase I was of no use to Services because it only contained quotes and clients. When the full version of phase I was installed it was not as useful as Q&A and this situation continued in phase II and had only begun to be addressed in phase III. The slowness of the system also impacted on Services because of the large number of screens that they had to use to process a single programme.

The problem with phase I and II was that they were designed from a management perspective but were not used because the data was incomplete. Underwriters were supposed to be entering some of the data but the system offered them little benefit. Services did use the system but found it less useful than Q&A.

f Data quality (Important - Method)

Without data quality the system was useless to management, for example in January 1995 they still could not use the system to report on their exposures to disasters such as the Japanese earthquake.

One of the reasons why the Claims aspects of the system were not used was because downloads could not match policy numbers because of the poor quality of the data in the Multinational system.

g Resource shortages (Important - Resources)

It is not clear that the resourcing implications of the introduction of the Multinational system were fully considered. By the time phase I was installed Multinational was understaffed and the introduction coincided with other major changes in the department. These staff shortages exacerbated the data entry problem and made it difficult to get people to attend training sessions. Moreover, the underwriters only had one PC between two of them, making it difficult to input data.

Installation in Services was complicated by the need to convert the data from the Q&A system. Services, who were already short staffed and with a backlog, spent time getting the data up to date in the Q&A system and then had a period of two weeks in mid June when they could not use any system.

h Installation (Important - Method)

The phased introduction of phase I was unfortunate because the first part of the system was of limited usefulness but it does seem to have been effective in introducing the technology to the users. It might have been better to have introduced Windows first and then had a proper launch of all aspects of phase I when they were ready.

i Aptitude of staff (Important - Individual)

There were mixed views about the importance of IT literacy or aptitude in getting a member of staff to use the system. There was a view that 'the further west you are the more IT illiterate you are' and that this explained why the UK and Holland used the system more than the USA. While there was some evidence that high IT literacy was useful, a critical factor was that staff needed to be prepared to persist and be interested in the system.

j Attitudes to change (Important - Change)

One of the reasons given for difficulties in getting the system used was that it involved change. Underwriters were 'sold the system' rather than having the implications of the change explained to them so they were unaware of the amount of data sorting out they would have to undertake. It is difficult to assess the importance of this factor but the increase in dissatisfaction in phase II caused by relatively minor changes indicates that it was important.

k Training (Important - Method)

Overall the team, perhaps encouraged by Tim, seem to have regarded prototyping and online help as substitutes for the more traditional approaches of training and user manuals. One of the features of phase III was a greater emphasis on training and user documentation. The relaunch in America was planned with an extensive training programme and training was included in the rollout to Europe. This was in response to a

recognition that there had been inadequate training in phase I particularly in America. The timing of training for phase I in the UK was inappropriate because of the phased implementation but there seems to have been a general lack of interest amongst underwriters or their managers for training in the UK.

There was no user manual produced for phase I because there was a belief, attributed to Tim, that none was required because of the development approach. Unfortunately there were also some problems with the online help system in phase I and America particularly missed the manual. The role of a user manual was recognised by the team but it can not have been regarded as very important as it took over six months to produce and I was never shown a copy.

1 Communication (Potentially important - Communication)

The development team made significant efforts to keep the users informed about developments. Despite this there were some negative comments about communication and one of the reasons cited for the failure of the electronic transfer system was a lack of understanding about how it worked.

7.3.2 Why was the quality of the data poor?

The responsibility for data quality seems to have resided with the users rather than the development team and in consequence the problems were not really highlighted by the Post Implementation Reviews. Moreover, some of the senior management of X seemed to have a relaxed attitude to data quality and this affected the underwriters' attitude.

There were difficulties in loading the data because of the way in which X had developed from a number of distinct groups working with different procedures and classification systems. More particularly some key information was only stored in underwriters' heads. Some of the difficulties in defining the data were inherent to the business of multinational insurance.

The data converted from the old Q&A system was already of poor quality and was out of date. In addition data problems were introduced at conversion by a lack of understanding

of the insurance business by development staff. This coupled with the decision to check the data at renewal meant that poor data quality in the UK persisted for over a year. Initially it was not possible for the users to correct some of the fields but when the validation rules were relaxed there was a further impact on data quality.

a Responsibility for data quality (Critical - Organisation)

In the Strategy Study in 1991 it was recognised that the new MIS system would depend on the quality of the data which was currently incomplete and that time would need to be spent sorting the data out. Despite this, issues of data quality do not appear to have been explicitly addressed in the early analysis.

Despite commitment to computerisation, senior management did not emphasize the importance of accurate data in the Q&A system perhaps because they did not have the same view of accuracy ('we don't deal in precise numbers, it's millions of pounds'). The loss of management commitment to the Multinational system when David left as phase I went live was regarded as one of the reasons the data quality was poor (although Jenny was temperamentally much better suited to getting the data right than David because of her Services background).

The problems of data quality were not really highlighted in the Post Implementation Review because it focussed on the development of the system. Even in early 1995 there was still confusion between the data quality problems and actually getting people to use the system. This perspective is reflected in the different ways the Post Implementation Review and Audit reports were handled. The Review was widely publicised but the Data Audit was kept under much tighter wraps.

In reporting the problems of data quality Colin attributed this to leadership problems but did not clearly identify who should have responsibility for it. The data audit was managed by the users so it would appear that they had responsibility for data quality at this stage. However, although the data audit found 'a lack of accuracy and attention to detail' and 'Mr Tim hit the desk over and over again', it took three months for X management to address the underwriters about the findings of the data audit by which time the impetus was lost.

It also took three months between Neville being given responsibility for sorting the data out and it being announced.

Throughout the development the team seems to have regarded data quality as the users' responsibility although they scheduled time to assist the users. As I stopped observing, the X Director responsible for IT, gave the responsibility for data quality to the development team. Whether in other projects the team would assume this responsibility is questionable but from their experience with X, they would certainly treat data quality problems much more seriously from the start.

While setting deadlines seemed to motivate the developers to delivering on time, deadlines set for underwriters to get their data input correctly do not seem to have the same effect. Was this because there was more of a project mentality amongst developers? I am not sure even the threat of unemployment will ensure Neville succeeds; in phase I giving Tim the ability to look at any underwriter's business may have 'frightened the life' out of the underwriters, but it did not provide any impetus to them getting their cases in and up to date.

b Attitude of underwriters (Important - Organisation)

The lack of interest shown by senior management in controlling the business meant that they did not emphasize the importance of accurate data in the Q&A system. Many of the underwriters had a similar attitude so the data to be converted from the Q&A system was already poor quality. There was some evidence that this attitude of the underwriters meant that 'any old thing was sometimes dumped in'.

c Organisation of data input (Critical - Organisation)

One of the problems of loading the data into the system was that the sources of information were not well organised. Some data was only stored in the underwriter's head, so Services who were used to keeping data up to date and regarded it as their job could not input it. There was a continued problem getting information from the overseas offices that was unrelated to IT, more 'hearts and minds stuff'.

The original plan was that the underwriters would input the full details of new cases because this would help them understand the system, they knew all the details of the case and it would give them ownership of data on the system. Unfortunately many of the underwriters in the UK and most of the underwriters in the US did not respond. There was a move to underwriters only entering programme level information for new cases. However, the files were kept in London and the new Services staff in Essex had little experience of the complexity of Multinational cases so this had not proved successful by the time I stopped observing.

The formation of X and its widespread nature meant that before the Multinational system was implemented there were not agreed common procedures. In addition, the decision to continue to take IT services (and other business services) from local X-Group operations meant that the MIS must take feeds from a great variety of uncontrolled sources who were 'organisationally and culturally distinct'. Tim and David saw the introduction of the new system as an opportunity to establish common procedures in X particularly for accounting. The difficulties of establishing common procedures were illustrated by the underwriter files; although they talked about standard forms for underwriter files in London in March 1994, these did not become established until August 1994 and one of the problems faced by the data audit team was the proliferation and variation of underwriters' paper files.

It was also recognised that world-wide aggregation of X data would require a common classification system to be established as part of the new system, but when data was loaded different interpretations were put on fields so they were used inconsistently. Field interpretation also depended upon business policy and so could change if policy changed.

d Difficulties in the data (Critical - Domain)

Getting the cases in was clearly not just an exercise in keyboard skills; even Ned who reckoned he was PC literate and thought he was up to date found a few errors when he was audited. The information was not all that precise, so for example the exact cover for a particular country might not be known without going back to the broker. There could also be a significant time lapse before the information was agreed. This complexity made it difficult to employ temporary staff to enter the data.

e Data conversion (Critical - Method)

Conversion of data from Q&A was responsible for many of the initial data quality problems. The data in Q&A was not of very high quality and was already out of date by the time the full version of phase I was installed. There were also a number of misunderstandings about how records should be converted resulting in problems with some programmes that were not apparent until the programme was individually checked. The problems of this conversion were so severe that errors were still emerging over a year later.

There was a view from developers and users that the data should not have been automatically converted from the Q&A system; at the very least they should have trialled the conversion to have identified the problems earlier. In phase II the data was converted again and further errors were introduced; some of these were caused by the poor structure of the phase I database.

f Development staff expertise (Important - IT profession)

A lack of understanding of the insurance business by development staff contributed to the problems in data conversion.

g Installation (Critical - Method)

The decision in the UK to convert the data from Q&A and then check it when it came up for renewal had a major impact on data quality. The consequence of this decision was that the rather fuzzy position on data quality was able to persist for well over a year. Moreover, Services had the burden of running two systems in parallel throughout that time.

It was definitely a bad decision to convert the data from Q&A with all its faults only to find that potentially it was not going to be checked for up to a year. Although the users still thought this approach was appropriate when viewed retrospectively, I feel that it would have been better to time the installation so that users would have been able to load the data immediately. This would have made it harder for data quality issues to persist. Holland took this approach (it was easier because they were smaller) and although their data still suffered some quality problems they seem to have used the system much more effectively.

h Validation (Important - Design)

The amount of validation on fields and the impact on data quality was an issue that continued to be debated. The amount of validation in phase I meant that data loading and correction was more time-consuming than it needed to be as some fields could not be amended by underwriters. In phase II some of the controls were relaxed allowing underwriters to save in the middle of a transaction but the data audit revealed that this led to more errors and a decision was taken to tighten up on validation.

7.4 DISCUSSION

This chapter has shown that both social and technical influences had an impact on how the software was conceived, constructed and delivered. Twelve key influences emerged from the categorisation of the events described in this chapter. The impact of these is shown in figure 7.1.

	Individual	Organisation	Method	IT profession	Belief	Resources	Domain	Technology	Design	Change	Vision	Communications	Social or Technical
Software conception or inception?													
Why build a system for Multinational?	★	★	✓				✓						S
What affected the choice of system boundary?	★	✓	✓					★					S/T
What affected the choice of systems architecture?	★	★	✓	★		✓	★	✓		✓			S
What affected the choice of process model?	★	★	★		★		✓						S
Software construction or evolution?													
What affected the functionality that was delivered?	★	✓	★		★	?	✓	✓					S
What affected the internal design?	★		✓	★		✓		✓					S
What affected the reliability of the system?	★	★	✓	✓	✓	✓	✓	?					S
What affected the schedules?	★	★	★	★		★		✓		?	✓	?	S
What affected the composition of the development group?	★	✓	✓	?		✓							S
Software delivery or use?													
What affected the use of the system?	★	★	✓		★	✓	★		★	✓		?	S
Why was the quality of the data poor?		★	★	✓					✓				S/T
Key													
	★	Critical		✓	Important		?	Potentially important					
		Social influence			Hybrid influence			Technical influence					
	S	Primarily social influences		S/T	Mixture of influences		T	Primarily technical influences					

Figure 7.1 Key influences on the software development in X

Figure 7.1 shows that there were three dominant key influences that were either critical or important in almost every aspect of the software development in X. Individuals had a critical effect on every aspect of development apart from the quality of data. Moreover it could be argued that the absence of an individual driving data quality was one of the critical reasons why it was so poor. The organisations involved in the development were critical or important in all aspects of the development other than the internal design of the

software. The methods adopted by the development team including the use of prototyping, project management and quality assurance procedures were important in all aspects of development and in some cases critical. In general, the methods had an important influence but were not as critically influential as individuals and organisations. This supports the impressions of chapter 6 that undertaking the development in a largely professional manner was not sufficient to ensure that the system was effectively used by all stakeholders.

Apart from the influences of individuals, organisations and methods, the development process was affected by six influences that had a major impact on some aspect of development and three influences that had a more limited impact. These three influences had a more limited impact because they were generally well-controlled at X. The development approach allowed for change, there was someone with a strong vision during the majority of the project and communications between stakeholders was generally good.

Writing the commentary it became clear that the course of the software development at X was affected by a large number of sometimes inter-related influences. Figure 7.1 illustrates this by showing that each aspect of development was affected by at least four of the key influences and even the less dominant influences affected several aspects of development across conception, construction and development.

An exception is design, which is only shown as influencing delivery. This is because the problems of the external design (functionality and usability) only became visible when the users started to use the system. We might have expected the poor internal design in phase I to have made construction more difficult in phase II but this was largely avoided by the rewrite and I did not observe long enough after phase II to see the impact of design on subsequent maintenance. It could be argued that the poor design in phase I was one of the reasons for the rewrite in phase II, but the impetus for the rewrite was largely a consequence of the prototyping methodology, and its scale was influenced by the loss of vision at the end of phase I and the nature of the IT profession. There was no real evidence that the reliability problems of phase I or phase II were caused by the poor design.

The key influences on the development in X are a mixture of social (individual, organisation, IT profession, belief, resources, change, communications), technical (method, technology, design) and hybrid (domain, vision) considerations as shown in figure 7.1. The final column in the table assesses whether the influences that were most important for any question were essentially social or technical. Overall this demonstrates that while technical influences were important, social influences were generally more critically important to the software development in X.

Section 8.1 explores each of the key influences identified in the commentary in greater depth by discussing the effects each had on the development at X and the potential for improving the development by understanding the influences better. The remainder of chapter 8 extends the understanding and utility of the key influences by comparing them with the findings in chapter 4 and discussing their effects on the other longitudinal cases.

8 Findings

Chapter 4 suggested that there might be a small number of key influences that can impact all software developments. One of the longitudinal cases, X, was selected for detailed analysis and chapters 5, 6 and 7 explain how the key influences on their software development were identified. This chapter explores these key influences (summarised in figure 7.1) and seeks to establish their wider utility by comparing them with the findings from chapter 4 and a summary analysis of the other five longitudinal cases.

Each event in the commentary on the software development in X (chapter 7) was categorised by the key underlying influence. The aim of this categorisation was to identify a small number of influences that were non-overlapping and non-contingent, and twelve key influences emerged. For each event, the impact of the key influence on the development was classified: **critical** if it made a significant difference; **important** if it had a clear influence but did not lead to major changes; and **potentially important** if it was not particularly important in Multinational but could easily have become important under different circumstances. In this chapter these key influences are discussed and refined as illustrated in figure 8.1.

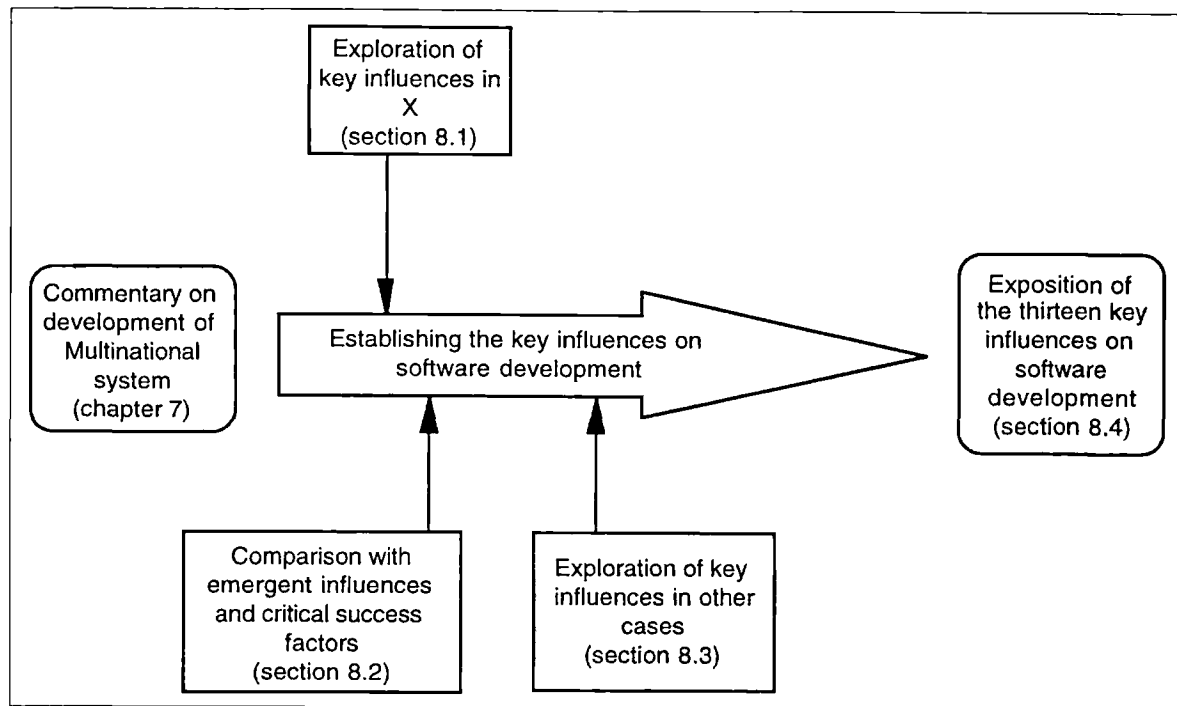


Figure 8.1 Establishing the key influences on software development

Section 8.1 characterises those elements of each key influence that affected the development. The utility of understanding these key influences is explored in X by discussing how the course of the Multinational development could have been affected by the participants giving explicit consideration to each key influence. The material in 8.1 is also contained on the cdrom where it is linked to the commentary in chapter 7 so that the evidence supporting the discussion presented in this chapter can be traced.

Section 8.2 compares the key influences in X with the emergent influences and critical success factors identified in chapter 4. Section 8.3 tests the applicability of the key influences in other cases by undertaking a summary analysis of the other five longitudinal cases.

As a consequence of the assessments in this chapter a thirteenth key influence is added and a number of questions are raised about software development in general. These are summarised in section 8.4.

8.1 THE MULTINATIONAL DEVELOPMENT

The commentary in chapter 7 reveals the complexity of the development. This complexity arose from the problem domain, the number of stakeholders involved and the time taken for the development. Multinational insurance is a complex business with no accepted definition of what it actually involves in detail. Each of the stakeholders had a potentially different view of what Multinational business is and what the objective was in writing a system. Over the time taken to develop the system, the nature of Multinational business changed, the stakeholders changed and the available technology changed. In writing the software these multiple, possibly incompatible views somehow had to be reconciled as only one system was developed. Added to this complexity is the difficulty of assessing what happened because this too is subject to the multiple realities of the different stakeholders varying over time.

Despite these difficulties, the concept of key influences affecting the development appears to be sound. This section examines the events coded to each influence to document how the key influences were significant. Aspects of each influence and the effects observed in Multinational are identified through a retrospective analysis of the events coded in chapter 7. Where the complexity of the events warranted it, this analysis was undertaken using NUD*IST. The results of this analysis is summarised as part of figure E.1 in appendix E and discussed in the following sections. These sections also consider the implications for the Multinational development, of a greater awareness of the key influences, by re-examining the commentary in chapter 7.

8.1.1 Individual

Software development is often regarded as a team activity but within X the talk always focused on named individuals. Individuals had a critical influence on the development of the Multinational system in four ways.

Specific individuals were a driving force behind the computerisation (7.1.1b) and had a vision for the scope and shape of the system (7.1.3b). The conception of the system was particularly affected by who was currently providing the vision for the system (7.1.2d).

The technical ability of developers was highly regarded in X and was critically important in the construction of Multinational (7.2.5a). The importance of appropriate experience was less widely acknowledged although experience (or the lack of it) had a critical influence in all aspects of development (7.1.4e, 7.2.3i, 7.3.1i). However, there were some occasions where an individual's lack of technical experience was unimportant (7.2.2d).

The personal qualities of an individual were critical in the construction of the system even though they were less often mentioned than ability. The personal quality most often mentioned was the ability to be tough or strong-willed (7.2.5b). Seniority of an individual sometimes had an important effect on the X development (7.2.3i) but was frequently not as important as an individual's commitment (7.3.1b).

It was recognised that the good personal relationship between Gordan and David was critical to the success (7.1.4e) but the problems of poor relationships were not acknowledged (7.2.5b). Leadership skills were important but were less often discussed (7.2.3i).

Implications

One of the key problems with the Multinational system was that phases I and II were of limited use to any group of users (7.3.1). The main reasons for this were that the scope of the system expanded to include operational issues but this was not made explicit (7.1.2a) and the actual content of the system was driven by the individuals involved (7.2.1). If they had accepted that an individual held the vision of the system then they might have taken more trouble to make the focus of the system explicit. For example, if the decision to replace the Q&A system had been highlighted then I think the team would have taken more trouble to ensure that the new system would replace all the useful functionality in the Q&A system. In addition, if they had been conscious of the impact of the specific people involved in the development on functionality, then they might have chosen different people. In particular it might have been better to second a practising underwriter rather than

Keith. This would have had the advantage of involving someone who was more aware of underwriter concerns and values.

When they were constructing the system I do not feel that Gordan took enough account of the strengths of his developers. He was very complimentary about their abilities while being realistic about their particular strengths and weaknesses (6.4.9). However he did not seem to allow sufficiently for the influence of the individuals on the quality of the product. For example, the lack of experience with the prototyping methodology should have alerted him to the danger of the project getting out of control as the management of prototyping is known to be difficult and had been flagged up at an early stage (6.4.8). Phase I was much bigger than planned in part because they had difficulty stopping the prototyping process (7.2.1a). A better understanding of the effect of the individual's lack of experience would have caused Gordan to manage the project more closely. Similarly, if George had taken more account of Kevin's lack of IT experience he might not have been appointed and this would probably have had a major impact on the problems of phase II (7.2.5a). Another area where respect for ability allowed the impact of inexperience to be ignored was in the design of phase I (7.2.2). One of the difficulties that Gordan would have had in addressing this is the difficulty there can be in managing IT staff (see discussion relating to the IT profession, 8.1.4).

The Multinational project began with a prototyping process model but by phase III had switched to a much more traditional development approach. Unfortunately, I did not observe for long enough to see the impact of this switch, however it is interesting that it was largely brought about through the influence of Jane and Colin (7.1.4g). Although Gordan highlighted the need for the organisation to learn from the problems of phase II, I do not think he supported the move (7.1.4.h). If he had been more aware of the impact of Jane's experience in this respect, I suspect that he may not have appointed her.

One aspect that was important in Jane's appointment was her ability to be 'tough' with senior management (7.2.5b). In general personality does not seem to have been a big issue in Multinational. Although there are no definite examples where an awareness of the personality of individuals would have been useful, I suspect an explicit consideration of

other personal qualities might have affected the appointment of Stuart to lead phase II (7.2.3i) and Jane's appointment (7.2.5b).

8.1.2 Organisation

Aspects of organisation were critical or important in almost all areas of the development at Multinational. The two main organisations involved in the development were the new company X and the developers, GIS, in X-Group. However development was also affected by the other companies in the group whose systems X was using (eg XUK) and to a lesser extent by the suppliers of the database package, Sybase. The organisations influenced the development in four ways.

The structure of X, both organisationally and physically influenced the conception and construction of Multinational. In particular the formation of X as a separate company (7.1.1c) was an important driver in its conception, and the subsequent growth and change in focus of X (7.1.2f, 7.2.5f) were important in raising the profile of the system. The ambitious decision to install the system in three countries in phase I was based on the structure of X (7.1.4c). Existing systems, both manual and IT, influenced the conception of Multinational by providing a driving force for a new system (7.1.1c, 7.1.3c) and delivery by being a barrier to the introduction of the system (7.3.2b).

Organisational culture was widely blamed for the difficulties in getting the underwriters to use the system. While this may not have been the most important reason for their non-use, organisational culture was critically influential in both the conception and delivery of the system. The most important aspects of this were the attitude to getting the business that was shared by many of the senior management and underwriters (7.1.1f, 7.3.1a) and senior management attitude to the accuracy of the data (7.3.2a). The difference in attitude displayed by Services to the system can partly be explained by the different focus of their work (7.3.1a).

Although there is some evidence of organisational learning having a beneficial influence during the development of the Multinational system (7.2.1i), there are more instances of the organisation failing to learn from early work (7.2.4e).

The relationship with an external organisation, the Sybase supplier, was critically important during the development (7.2.3f).

Implications

Understanding the impact of the organisation on development can have less direct impact on how projects are conducted than an understanding of the influence of individuals because it is rarely possible to change the organisation. Nevertheless by understanding the impact, developers can either accept the organisational influences or make greater efforts to overcome them.

The decision to install phase I in three countries simultaneously was driven by business needs rather than methodological considerations (7.1.4). Given that America were so unsuccessful at using the system it had to be relaunched eighteen months later (6.8.7) and the size of the system put a strain on the development, it would obviously have been better to have installed in America later. I think an understanding of the influences of the organisation would have allowed the developers to consider more carefully the choice of countries for initial installation. Furthermore an awareness of the impact of organisational culture would have enabled them to predict the installation problems in America and so leave its installation to later.

One of the other major problems of the system, the update and quality of the data, would have been eased by more account being taken of organisational culture (7.3.1). The IT Strategy highlighted that there would be difficulties in persuading the underwriters to update their data (7.3.1a) but this was not taken seriously enough by the developers. The development of the system was used to bring about organisational change by unifying the systems from the different companies that formed X (7.3.2c). A recognition of the importance of existing systems may have persuaded the development team to spend more time considering these systems and deciding how to bring about the unification of the procedures and classification systems.

8.1.3 Method

The method had an important influence in all aspects of the development of Multinational and at times was critical. Sometimes the impact of the method was positive, sometimes negative and sometimes it had no impact or was ignored.

The only positive critical influence of method was the use of prototyping to build a large system in phase I on time and under budget (7.2.1a). However prototyping and the requirements study also had important positive influences on the development (7.1.3a, 7.3.1c) and on the careers of both users and developers (7.2.5d).

Despite its positive influences, prototyping also had both critical and important negative influences on both construction (7.2.1a, 7.2.2a) and delivery (7.3.1k) and the requirements study also had a critical negative influence by specifying a first phase that was too large (7.1.4b). Project management, installation and data conversion also had negative impacts on development (7.2.4a, 7.3.1h, 7.3.2e). These negative influences were not just caused by tasks being undertaken badly or inappropriately but also because social influences overwhelmed method.

During development some appropriate methods that were used were either ineffective or ignored resulting in critical or important impacts on development. Project management and the IT Strategy both come in this category (7.2.4a, 7.1.2a). Quality Assurance was also largely ineffective or ignored but the impact was rarely important or critical (7.1.2e, 7.2.3e).

Implications

The development of the Multinational system is interesting because methodologically they did most things well but still had significant problems with the development (6.9.4). There are a number of ways in which they could have improved the development approach or at least made it more effective by recognising the limitations of method in affecting the course of software development.

There was a view that the IT Strategy was only undertaken to justify a system which Tim had already decided to implement (7.1.1) and Colin had already designed the architecture (7.1.3b). Maybe it was this or a more general industry attitude to IT strategies (6.8.8) but

the results of the IT Strategy were largely ignored (eg replacement of Q&A system, difficulty of getting underwriters to use the system). It would have been better if the IT Strategy had been heeded or failing this maybe time and money could have been saved by doing a reduced version to satisfy X-Group.

The quality assurance functions in GIS were also largely ignored by the developers (7.2.3e, 7.2.5e). This was partly because they were out of step with the prototyping approach adopted by the team. Attention to the Post Implementation Review for phase I might have resulted in the leadership issues of phase II being addressed quicker but the worst consequence of the lack of impact of the quality assurance function is that it wasted time and money.

The use of prototyping had a large impact on the development of the Multinational system, both positively and negatively. A better understanding by the team that method can impact on functionality (7.2.1a), design (7.2.2a), testing and thus reliability (7.2.3c) and scheduling (7.2.4a) would have led them to view prototyping more critically. This might have enabled them to gain the benefits from prototyping without suffering so many of the drawbacks that they had to abandon it in favour of more structured approaches.

8.1.4 IT profession

There were some personal characteristics that had a critical influence on the Multinational development. These are categorised to 'IT profession' rather than 'Individual' because they appear to be shared by many in the IT profession.

The conception of the system architecture was influenced by a preference amongst the developers for a particular technology (7.1.3e). The remaining critical influences were in system construction where a very wide range of skills was required from the IT staff but their interests did not always coincide with what was expected of them (7.2.2e). The developers consistently under-estimated what had to be achieved (7.2.4f) and there was some evidence of 'second system syndrome' (7.2.4a). During construction the difficulty of effectively managing IT staff (7.2.2f) and their view of success (7.2.3b) were important

influences. The project was affected by difficulties in recruitment and retention of IT staff but this did not have an important impact on the project (7.2.5h).

Implications

An understanding and acceptance of these character traits would have helped Gordan and the other project leaders to allow for this in the development.

One of the unfortunate decisions during the Requirements study was the decision to use PCs for the servers rather than Unix (6.9.1). This decision was partly influenced by the increasing power of PC technology but I think the resistance to Unix by some of the developers probably also had an effect (7.1.3e). Being aware that IT staff preferences may not be matched by technical considerations might have convinced Gordan and David to stick with the original plans and thus avoid the Sybase problems.

Similarly an awareness of the over-optimism of his IT staff might have allowed Gordan to schedule more effectively (7.2.4f). IT staff need to have a very wide range of abilities, both business and technical, to do their job and this was generally recognised in the Multinational development (although on occasions there were problems because of lack of ability eg 7.3.2f). However, one of the reasons for the problems of phase II was a failure to realise that IT staff may like the challenge of new design work but may not be 'completer/finishers' (7.2.2e).

The Multinational development also illustrated how difficult it was to manage the IT staff and really know what they were doing (7.2.2f, 7.2.3h). Recognition of this difficulty might have resulted in Gordan being involved in the decision to rewrite phase II.

8.1.5 Belief

The success of the Multinational system was affected by the commitment and enthusiasm of key individuals but was also critically influenced by a more general belief in the system. In the context of the developers this belief in what they were doing meant that they achieved a lot in a short period of time (7.1.4d) but also led to the system being larger than

~ -

expected (7.2.1c). Amongst the users, maintaining a belief in the system had a critical impact on how they regarded and used the system (7.2.3a, 7.3.1d).

Implications

Realising that the developers' belief in the system could impact the development process might have alerted Gordan and David to the growth in the size of the system (7.2.1c). An awareness of the importance of belief in the system would certainly have allowed them to realise that America was unlikely to be successful while senior management did not believe in the system (7.3.1d).

8.1.6 Resources

Resource limitations were an important but rarely critical influence on the development. With staff the limitation was not that there were not enough staff but that existing staff were too busy (7.2.3d, 7.3.1g) and it took a long time to recruit or release new staff (7.2.5g). This delay seems to have been due to the time required for recruitment rather than difficulties in recruitment (7.2.4d). Time constraints occurred because of the need to meet deadlines (7.2.2c) and the timing of those deadlines (7.2.4d). Financial constraints were more concerned with budgetary processes rather than particular shortages of money (7.2.5g).

Implications

Resource limitations were not really a problem for Multinational but nevertheless an understanding of the impact of resource shortages would have helped them to make more effective plans. One of the notable features was the time taken to recruit project leaders and the loss of momentum of projects during this period (7.2.4d). Although it might not have been possible to reduce the time, Gordan could have taken action to progress the project had he been aware of the delay at the outset.

The delivery of phase I was affected by a shortage of staff in user departments to undertake the data entry (7.3.1g). This problem had been signalled in the IT Strategy but insufficient allowance was made. If they had been more aware of how important it was to consider

staff shortages, they could have either re-timed the delivery, recruited additional staff or allowed for overtime.

8.1.7 Domain

Software development literature takes account of the problem domain at the macro level; for example, there are differences between development approaches for real-time and persistent data systems. This key influence illustrates how lower-level characteristics in the problem domain (eg complexity, external factors, absence of an existing system) had an important and sometimes critical influence on the development (7.1.3g, 7.1.1e, 7.3.2d).

Implications

If the developers had more carefully identified the complex aspects of the domain such as reinsurance (7.2.1e), they would have improved development by using developers with the right abilities. If they had had a broader understanding of the nature of the Multinational data (7.3.2), they would have been in a better position to design a system to handle such data and would not have been content with the rolling conversion of UK data which allowed inaccuracies to persist.

8.1.8 Technology

The capabilities of the technologies (hardware, networks and software) used in Multinational were important drivers and constraints on the development (7.1.3d, 7.2.1d, 7.2.4g). The availability of a working system was also critical in enabling the users to see the limitations of the system for Services (7.1.2c).

Implications

In drawing up the initial project schedules, allowance was made for the team's inexperience with the technology as a whole. However they did not allow for the technology making the task more difficult than anticipated (7.2.4g). If the technology used in the Financial Accounting sub-system and ad hoc reporting had been regarded more cautiously they would

have been able to build time into the schedule for experimenting and getting a measure of its capabilities.

8.1.9 Design

The result of design was critically influential in developing a system that was usable and useful to the various groups of users (7.3.1e). Despite this, users or developers rarely explicitly discussed design.

Implications

One of the striking things about the Multinational system is how little use phase I or II was to the majority of the users although no-one made this explicit (7.3.1e). This reflects the invisibility of the design process at Multinational. Throughout my observation, I was struck by how little they talked about design (external or internal) and how rarely they showed me design documents. The prototyping process in phase I seemed to focus on the minutiae rather than the bigger picture. I believe that had they made design more visible they would have been clearer about the system aims (and so would have built a more useful system) and may have designed a better internal system in phase I.

8.1.10 Change

There were two types of change that had potential for influencing the Multinational development; external changes affecting the development process and changes caused by the development itself. Both types of change were important, but not widely discussed. Change seems to have been considered more fully in technical design than in its influence on people (7.1.3i, 7.3.1j).

Implications

At one time change was recognised as part of software delivery but the pace of change today can mean that the process of change is not taken seriously. I think this, possibly encouraged by the use of prototyping, is what happened with the delivery of phase I of the Multinational system (7.3.1). Just simply accepting that change needed to be managed

would probably have meant that the system was adopted more successfully amongst underwriters in the UK and America.

8.1.11 Vision

The vision of Multinational was held by a number of individuals during the period of observation. The individual holding the vision was very influential as already discussed but the loss of momentum that occurred when the vision was lost shows how important it was that at least some body was providing the vision (7.2.4c).

Implications

If they had been more aware of the importance of the system vision in driving the development forward then they might have been better prepared to handle the loss of vision at the end of phase I (7.2.4c).

8.1.12 Communication

Communication is normally regarded as a key difficulty in software development but was not identified as such in the Multinational development probably because of the success of the team in communicating with stakeholders. However, the small number of reported incidents do illustrate how easily communication failures can have a significant influence on development (7.2.4h, 7.3.1l)

Implications

Given the lack of real communication difficulties during the Multinational development it is hardly surprising that no major improvements in this area can be identified.

8.1.13 Reflections

This section shows that knowledge of the key influences could have improved the Multinational development, making it more useful, reliable and efficient. However, the intrinsic complexity of the development means that complete success is impossible. The

remaining sections in this chapter examine how far these findings apply to other software developments.

8.2 COMPARISON WITH RESULTS FROM CHAPTER 4

In chapter 4, eight potential influences were described emerging during data collection for all cases (section 4.2.6) and nine critical success factors for object oriented development arose from the think tank (section 4.3.2). At the think tank fourteen factors affecting software development were identified and nine were selected for discussion on the basis of votes from the think tank participants (see section 4.3.1). This section discusses the potential for generalising the findings by comparing these results with the key influences discussed in section 8.1.

The results from chapter 4 are mapped against the key influences in X in figure 8.2.

Key influences	Potential influences (4.2.6)	Think tank CSFs (4.3.2)
Individual	Power of the individual	
Organisation	Influence of the organisation	Culture
Method		Management and control Process and planning
IT Profession		
Belief	Need for vision and belief	Investment/Commitment
Resources		
Domain		
Technology	Problems of technology	Technology
Design	Importance of design	Design
Change	Essential ingredient change	
Vision	Need for vision and belief	Technical vision
Communication	Difficulty of communication	Communication
	Impact of chance	
		Team

Figure 8.2 Comparison of key influences from analysis of X and results from chapter 4

As might be expected, the potential influences in chapter 4 that emerged from data collection of all cases map quite closely on to the key influences. The analysis of X provided more examples of the importance of each factor and introduced new factors. However, there is also quite a close correspondence with the think tank critical success factors although there is a greater emphasis on the more technical factors. This probably reflects the beliefs of a largely technical audience. There is more overlap between the think tank factors but this is because of the speed with which the categorisation was done. The main differences between the three sets of influences are:

- The think tank did not identify 'Individual' influence but instead had a 'Team' factor. This is hardly surprising given the emphasis in the literature on the team and the think tank did identify 'Individual' as a factor with nine votes. Moreover the 'Team' factor did include some of the issues covered in the 'Individual' influence (individual role, skills and leadership).
- The only think tank factor which was concerned with 'Organisation' was 'Culture' and this was mostly concerned with the culture in the development team not the culture in the organisation using the software. Given the critical effect of the structure and culture of X on all aspects of software development it is surprising that the 'Organisation' did not feature in any of the think tank factors but perhaps reflects the participants' largely technical viewpoint. Given the nature of the conference it is likely that they had the software engineering (see 2.2.1) or project management view (see 2.2.3) of software development and these views do not normally see the organisation as an important feature in software development.
- For much of the analysis of X, the influence of 'Method' was coded under 'Design' but this was divided to emphasise that there are two distinct activities: the process and the product. 'Method' describes the effect of the explicit software development process including project management and quality assurance activities on the conception, construction and delivery of software. In the think tank, these activities were divided into two factors reflecting I suspect the interests of the participants in methods. In contrast, 'Design' describes the influence of the product (in the broadest sense) on the software development. The difficulty of differentiating these concepts is reflected in the think tank 'Design' factor, which includes elements of process and product.
- Three new influences (IT Profession, Resources, Domain) emerged during the analysis of X. The 'IT Profession' and 'Domain' factors were not explicitly discussed in conversations but were identified from an interpretation of attitudes and events. This is probably why they were not identified during the think tank although aspects of domain were in the 'Magnitude & scope' factor which only got six votes.

'Resources' is a more obvious factor and elements of this were included in the think tank method factors.

- The potential influences in chapter 4 combined 'Vision' and 'Belief' but both the analysis of X and the think tank recognised that these were two different things. 'Vision' is about one person (the think tank 'messiah') having a view on where the development is heading. This vision needs to be shared but is essentially held by one person, which is why much of the importance of vision in the X analysis is coded to 'Individual'. In contrast, 'Belief' is about a group of people believing in what they are doing. In the think tank the emphasis was on the commitment of users but in X the belief in the system by developers was also seen to be influential.
- Chance was not identified in the analysis of X or the think tank where it received two votes. This is discussed further in the analysis of the other cases in section 8.3.

The correspondence between the influences is supportive of the generalisation of the key influences but raises a number of issues, which will be further discussed in section 8.4.

8.3 ANALYSIS OF THE OTHER CASES

To evaluate the applicability of the key influences in X, the other cases studied and described in chapter 4 were analysed to determine the main issues affecting the developments.

This was a much briefer analysis than the analysis of X and was undertaken after the analysis of X was completed and section 8.1 written. The documents and notes for each case were read and salient passages extracted to form a brief chronology of the development and issues affecting it. The issues were then categorised according to the key influences in X and their importance assessed using the same criteria for 'critical' and 'important' as in X (see start of chapter 7 for a full definition). Given the limited analysis and the tight integration between the two projects at Y, these two projects were analysed together. The effects of the key influences in each organisation are summarised in figure E.1 in appendix E.

The analysis of these case studies was limited for a number of reasons. At V the interviews were mostly retrospective and so tended to rationalise events. Moreover my insider status

meant that the participants may have been reluctant to criticise those aspects of the project in which I was involved. This is one reason why no design influence is recorded in this case. The data collected for W was far less detailed than for the other cases because of its original status as an experiment. The analysis of Y and Z were more comparable to X but the linear analysis of the notes of the interviews meant that it was generally only possible to identify issues that the participants alluded to directly and it was not always easy to assess whether a view was widespread. As I did not interview any of Y's customers, it was not possible to get the user perspective in this case. Z was such a large project that it was difficult to do it justice in a summary analysis. Moreover as I did not follow Z in detail through to implementation, influences during delivery were not generally identified. In particular it was impossible to really assess the product and thus the design influence.

Despite the difficulties with the analysis, the results (summarised in figure 8.3) indicate that the other cases exhibited the same key influences; 'chance' has been included in the list of influences because further analysis revealed that it was still important for some cases. Although the nature of the analysis was such that minor influences may not have been detected, no other new important influences emerged.

Key influences	X	V	W	Y	Z
Individual	★	★	★	★	★
Organisation	★	★	★	★	★
Method	★	★	✓	✓	★
IT Profession	★			★	★
Belief	★	★		★	✓
Resources	★	★	★	✓	✓
Domain	★	✓		★	★
Technology	★	★		★	✓
Design	★			✓	✓
Change	✓				★
Vision	✓	★			✓
Communication	?	✓			★
Chance			★	★	✓

Figure 8.3 Impact of key influences from all cases
 (★ - critical, ✓ - important, ? - potentially important)

Figure 8.3 shows that the key influences generally had a similar impact in the other cases. Case W, the small specialist joinery company was less affected by the key influences than any of the other cases but this was because of the nature of the development and the limitations in the data collected.

In figure 8.3 the influences are listed according to how much impact they had on the development in X (see figure 7.1). In X, three influences (individual, organisation and method) were critical or important throughout development with method often less critical than individual or organisation. The other cases confirmed the overwhelming importance of these influences and supported the finding that method was important but not critical.

In X the next six influences in figure 8.3 were identified (see section 7.4) as being more influential than the remaining three influences ('chance' was not identified as an influence in X). This distinction is not supported by the analysis of the other cases. Z was influenced critically by change and communication and V critically by a loss of vision.

Figure E.1 in appendix E shows that within a key influence there was broad agreement on the effects. The vision, ability, experiences and personal qualities of **individual** developers and users were important in most cases. For example the way in which technical ability was valued in X was very similar to how Oscar was forgiven for his print server problems in Y because his 'skills are valuable and rare'. Relationships were only identified as important in X but as these were rarely discussed in X, it is perhaps not surprising that relationship issues were not identified during the more summary analysis of the other cases. Although the software was developed in teams in X, V and Z, the team did not feature as an important issue in either case. This may have been because the teams were working well but this was certainly not true in V. This emphasis on the individual reflects the way in which each part of the software was actually built in X, V and Z by individuals or in the case of Z sometimes a 'teamlet' of a user and a developer.

The structure and culture of the **organisation** were critical in most cases and for quite similar reasons. For example the culture and differences in attitudes between underwriters and Services in X is mirrored in Z by the differences between academics and administrators. In both cases these differences reflect the groups' different goals within the organisation. Relationships with external organisations were important in three cases but the ability of the organisation to learn was not identified as important in any of the other cases. In part this may have been because observation did not cover a complete cycle of development in the way that it did in X.

Method was a major issue in X and Z. Although both used a user-centred approach with users working on the project team, initially they were at opposite ends of the informal/formal spectrum. Method was rarely discussed in conversations with V (probably because of my insider status), W was too small for method to be much of an issue, although PQM was used, and Y did not use an explicit method. Both X and Z had issues in similar areas and these were generally unremarkable in industry terms. In both cases the IT strategy was perceived as important but its use did not seem to warrant the effort that was expended on it. Z invested a lot of time and money on establishing and developing its methodology initially while X invested this effort when it switched to a more structured methodology in phase III. In both X and Z the methodology appears to have had mixed

success and it is not clear that the investments in methodology were worthwhile. Installation was not really observed at Z but there were data problems in both cases. Project management was more strictly applied in Z but the results indicate that to a certain extent this only provided an illusion of control. In both X and Z, quality assurance was in place but there were doubts about it having any impact on system quality.

Of the other influences the following similarities can be seen from an examination of figure E.1:

- Characteristics of the **IT profession** were clearly identified in X, Y and Z (summed up by a joking remark by one of the users in a project meeting at Z referring to the IT staff as the 'scum of the earth') and in each case difficulties of managing IT staff were identified.
- Maintaining the **belief** of the users in the system is well understood and is often part of communication (called 'PR' by Jenny in X), but the effects of maintaining the belief of developers was also important in X, V, Y and Z. This was illustrated by the dramatic change in progress once Oscar became convinced that he could write the print server.
- **Resources** issues that were identified as important in four of the cases was the impact of delay in recruiting or seconding staff.
- External factors, that were an intrinsic part of the **domain** and therefore not under the control of the developers, were important or critical in X, Y and Z.
- Changing **technology** was important in X, V, Y and Z. V illustrated the same kinds of issues as X of new development tools being both more powerful but having limitations. In X the tools were subject to careful selection and evaluation while in V the team had no choice of tool, so it would seem that this is likely to be a feature of any software development that builds upon sophisticated tools.
- The only **design** that was visible in X, Y and Z was functionality despite efforts during conversations to discuss internal design. (In V, I was very conscious of the internal design because of my involvement and although I could see problems with how the design had been continued since my departure, this was an area that was difficult to get the participants to talk about.) Although this lack of visibility caused problems in X and in Y it was too soon in the development cycle to see the impact in Z.
- Much of the **change** in X that was important in the software development, such as X becoming a separate company and the impact of existing systems, was categorised as organisational issues. The change category was reserved for two specific types of change: external changes that were not part of the development such as the changes

in the office that necessitated recabling, and resistance to system induced change. This decision reflects the way in which the system at X brought about change but this was not made explicit there. The situation in V was very similar but in Z, project C was seen as an agent for change and handling the change was an integral part of the development process. In consequence, change was perceived to have a more critical impact on development in Z.

- As already discussed in 8.3.1, much of the importance of **vision** was categorised under individual reflecting the importance of who was actually providing the drive and vision. The same situation occurred in V, Y and Z.
- **Communication** was not seen as a significant issue in X probably because their communication strategies generally worked well. Z suffered problems of communication because of the size of the project despite huge efforts to communicate with the rest of the University.
- The influences of **chance** that were identified in W, Y and Z are events that could not be realistically planned or controlled in a project. Knowing that there are things that are outside control allows a project to adapt and utilise chance events. Paul (in Z) used the analogy of a good canoeist who makes the most of the unseen rocks and currents. For this reason, chance has been added to the list of key influences.

8.4 DISCUSSION

This chapter has identified and described thirteen key influences that can have a critical effect on software development. These influences are not necessarily things that the developers can control but by understanding the impact of the key influences, developers can improve software development as illustrated for X in section 8.1. Twelve of the key influences emerged from the analysis of X and the thirteenth, chance, was added from a consideration of the other cases. A comparison with the results of chapter 4 and a summary analysis of the other cases indicates that these influences are likely to be generally applicable. Three influences (individual, organisation and method) can be regarded as dominant influences that will affect most developments and certainly all substantial developments. Large developments will also be affected by many of the other influences. While the research has not demonstrated that there are no other key influences, the thoroughness of the research method indicates that there are no other dominant key influences.

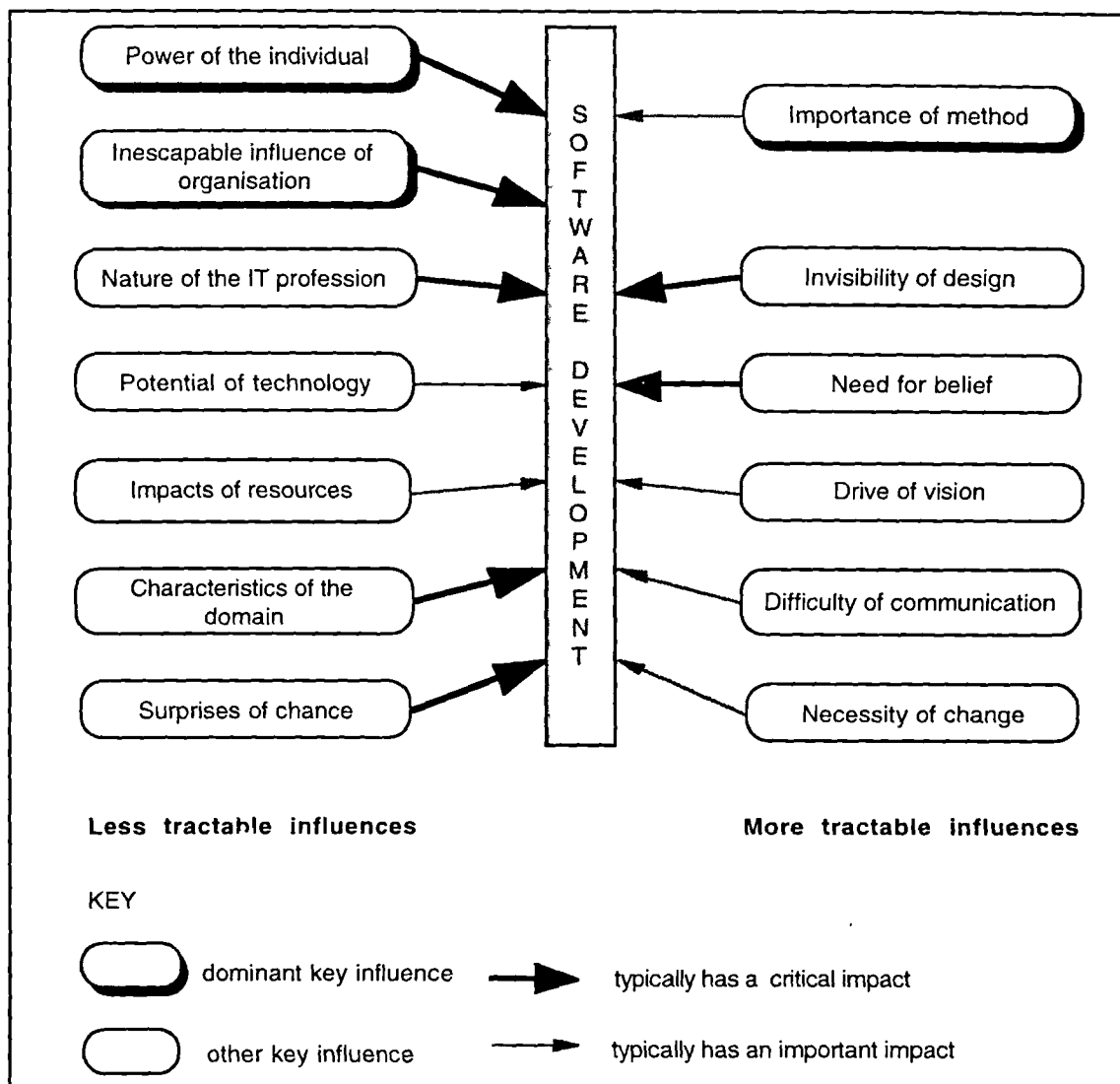


Figure 8.4 Key influences on software development

The key influences need to be considered from three standpoints: the different aspects of the influence that are important, typical impacts of the influence on development and the implications for the practice of software development. Further research needs to be done to exhaustively address these areas but the findings from this research are summarised in figure 8.4. These key influences provide those involved with software development with a series of influences to seriously consider, rather than a set of prescriptive actions. The phrases in figure 8.4 encapsulate the important characteristics about each influence revealed by this research and are summarised in the following discussion about each influence.

8.4.1 Key influences

Individual

Each person involved in the development can be very influential whatever their seniority or role. Important characteristics to consider are: the perspective of the individuals providing the drive and vision; ability, which is useful but not sufficient; experience, which may be very important but is not always necessary; and personal qualities and relationships.

Organisation

The structure, systems and culture of each organisation involved in development (including related external organisations) can have a critical effect. Although the organisational context can not normally be changed, an understanding of it can enable developers to make better allowances for its effect.

Method

Software development methods (in the broadest sense) are important but developers need to be more aware of their strengths and weaknesses. It is also very important to realise that a good method may not have the desired result because of the impact of other key influences that can often be more critical.

IT Profession

An understanding and acceptance of the character traits of many IT staff can help those leading developments to take account of those traits and thus manage them more effectively.

Belief

Belief of developers in the system can have a critical effect on their productivity and belief of users can critically affect their use of the system.

Resources

Shortages of resources do not just have an impact on schedules they can affect other facets of development such as project morale or successful adoption. Budgetary concerns can impact on technical decisions in ways that are not made explicit.

Domain

While the domain of the problem is a given, it is important to understand those aspects of the domain, such as areas of complexity, for which special allowance needs to be made.

Technology

The technology being used can make a huge difference to the ease or difficulty of a task and this is not always obvious until it is actually used, despite good evaluation exercises.

Design

Aspects of the design of the product (both internal and external) are often not made explicit and these can hide problems.

Change

Although change is commonplace it still needs to be managed in construction and delivery.

Vision

The system vision is an important driving force for development.

Communication

Communication is important and difficult to get completely right.

Chance

A development approach needs to be flexible enough to accommodate chance events.

8.4.2 Consequential questions

Each of the influences described above merits further research and many will be recognised as important from experience by software developers. However, comparison with the results from the think tank and the discussions in this chapter indicate some surprising results, summarised in the following questions:

- Do developers focus on things they can control at the expense of considering other more important influences?
- Do teams or individuals build software systems?
- What are the impacts of organisations on software development?

- Do methods focus too much on process at the expense of product?
- What are the implications of the multiple realities of software development?

The implications, for practice and research, of the key influences and these questions are addressed in chapter 9.

9 Conclusions

This conclusion discusses both contributions of the research: a research method for longitudinal case studies and key influences on software development. For each contribution this chapter discusses the work undertaken, the findings and the implications for practice and research.

The initial motivation for this research was to increase understanding of the processes involved in developing software. Building on Brooks' (1987) view that there are inherent properties of software that make it difficult to build, this research adopted a broad definition of software development. This was studied as a human activity in order to expose new ideas by looking at the process through, for a computer scientist, an unusual lens. With a strong computer science background I joined a Management School to help this refocussing.

The decision to study software development as a human activity was justified by a survey showing that both technical and non-technical participants felt that software development was affected by a variety of largely social factors (chapter 2). Despite this the literature review (chapter 2) showed that social influences on software development are generally treated in a fragmentary way by the literature. In part this fragmentation arises because the literature spans the disciplines of computer science and information systems. Five separate views of software development were identified from an analysis of the literature, each emphasising different aspects of human influences.

The survey in chapter 2 revealed the inadequacies of survey methods for addressing the research topic but this characterised the prevalent positivist approach, at the time, in the contributing disciplines of computer science, information systems and management. The most important outcome of chapter 3 was the decision to adopt research approaches from

the social sciences and education. This meant I had to accommodate a paradigm shift and that the method of research was not clear-cut. The implications of adopting a constructivist-based paradigm were far-reaching for both this research and my personal development. One consequence of this was that the research question was reframed in chapter 3 and refined in chapters 4 and 5.

The exploratory nature of the research question led to the decision to study six software developments in five organisations longitudinally over approximately two and a half years. Longitudinal studies were used to address the 'plausibility bias' in cross-sectional studies where participants reconstruct plausible explanations for historical events (Curtis, Krasner & Iscoe 1988). A pilot study was undertaken in organisation W but the development was too small to illustrate the range of problems that occur in software developments and, as described above, the research paradigm changed. Within the limitations of a PhD it was not feasible to follow more than five further developments in the depth required by the research paradigm, and, for the time necessary to observe a sizeable software development through phases of conception, construction and delivery.

A large quantity of qualitative data was collected ethnographically primarily from taped open-ended conversations and project documentation (chapter 4). Initial exploration of the longitudinal cases led to the idea that there are a number of fundamental or key influences on any software development process (chapter 4). The remainder of the research was devoted to determining what these key influences might be. In line with the research paradigm, the aim was to identify and understand the influences rather than model cause and effect.

The approach taken was to study one of the cases, X, in detail using the research method presented in chapter 5. The design of this method arose from the need to develop a method of understanding longitudinal cases that would be believed by the target audience for the research but was consistent with the new paradigm. One of the guiding principles was that the method should meet requirements for research quality. In keeping with the paradigm shift and the needs of my audience, I defined criteria based on goodness and utility (section 3.4). These criteria build on the positivist criteria of reliability, objectivity and validity in

order to make the research acceptable in disciplines dominated by positivism and have been successfully used to defend exploratory research (Oram 1998). Section 9.1 describes how the research method was designed to meet these criteria and explains the need to develop a hyperlinked presentation of the results (contained on the accompanying CDROM). The implications for qualitative research and software supporting qualitative research are considered.

Chapters 6 and 7 and the CDROM contain the results of a detailed investigation of the key influences on one software development. The findings are discussed in chapter 8 and extended with a summary analysis of the other five longitudinal cases. A case is made for the wider relevance of the thirteen key influences. This contribution is assessed in section 9.2. Section 9.3 briefly discusses the wider effects of the choice of research paradigm.

9.1 EVALUATION OF RESEARCH APPROACH TO LONGITUDINAL CASE STUDIES

Kvale (1996) devotes an entire chapter to addressing the '1000-page question'. It is a question that I often meet in teaching the use of qualitative research software, and it is a question, rephrased, that I faced in this research:

'How shall I find a method to analyze the 2387 pages of documents and notes and 102 hours of taped conversations I have collected?' (adapted from Kvale 1996 p176)

Like Kvale, my first reaction when I am asked this question is to be horrified that a researcher has got to this position. Accepted wisdom in qualitative research is that analysis should proceed in parallel with data collection, that this is too much data to manage qualitatively even with computer support, and before data collection begins you should have some idea of method. However, having been there, this is easy to say but not realistic for a researcher conducting their first piece of qualitative research in a discipline without strong traditions of qualitative research.

This problem was addressed by realising that, like most researchers, analysis had been taking place while collecting data leading to the concept of key influences (chapter 4). Treating one case in detail and using the other cases to support arguments of utility

(chapter 8) mitigated the dilemma of too much data. Solving the problem of how to analyse one case in detail led to the method described in chapter 5, which was developed during the four years it took to analyse X.

9.1.1 Development of research method and associated CDROM

In order to meet criteria for research goodness the research method needed to be acceptable to those involved in software development (specialist developers, users and managers) so that they could judge the credibility of the results. The technical people in this audience are likely to have had a scientific training and so will respond to a method that is detailed and carefully explained without large steps of interpretation. For some in this audience, perhaps especially those who are non-technical, the results may be surprising and so they will need rich detail to convince them of the credibility. To meet the utility criterion it is necessary for the reader of the research to know enough of the richness of the case to be able to transfer the results to other situations.

These requirements led to the design of a research method that allows for a clear audit trail from the results to evidence in the data. Presenting the results in a CDROM hyperlinked to the original data allows the reader to follow this audit trail. It also provides access to the richness of the case without overwhelming the reader with detail.

The research method (summarised in figure 5.1) uses two novel techniques:

- **Construction of chronological story told in the words of the participants**
This is achieved by dividing the data (interview transcripts, notes and documents) into evidence extracts and ordering the extracts so that all those referring to the same event (retrospectively, contemporaneously or in the future) are together. The evidence is then trimmed to remove redundancy and make the conversational text easier to read. This technique allows the multiple realities that are reflected in how people talk about events to be revealed.
- **Multi-tier analysis of chronology to identify findings**
This involves categorising the text in a number of dimensions, writing memos about the categorisation and then treating the memos as data. This technique allows abstraction to be developed in a visible manner.

The software developed in Hypercard as part of this research (section 5.2) supported both these techniques. This was essential as even now no commercial software provides all the capability required and the amount of categorisation and reordering would not have been feasible using manual techniques.

The software provides some limited capabilities to allow the development of the findings to be traced back to the source documents. Its use was considered for the hyperlinked presentation of the results but web technology was adopted primarily because of the portability that results from using a restricted subset of HTML (section 5.3). The text for thesis chapters 6, 7 and 8.1 and all but two of the 1600 web pages was generated by programs written in Hypercard. This ensures that the analysis, thesis and CDROM are consistent. The program generates the necessary HTML to provide the menu, navigational and hyper links incorporated in the web pages on the CDROM. Although web technology has been used to represent qualitative data, none of the commercial programs support this level of integration of analysis, text and web.

9.1.2 Evaluation of research method

The strengths of the research method are:

- provides a technique for explicitly comparing multiple realities;
- the technique of multi-tier analysis is useful for those who have difficulty in achieving distance from the data and where a detailed audit trail is desirable;
- the web is a good medium for representing hyperlinks and makes research accessible to technical audiences.

However, the method suffers from the following drawbacks:

- time taken to develop the results – although this is true of most qualitative approaches it is aggravated in this method by the need to keep a trail of results;
- loss of variety of data and technique – a well-known problem when using software to support qualitative approaches;
- no commercially available software supports the technique of story construction or web page generation so it would be difficult for others to use.

9.1.3 Implications

The use of dimensions to support the category development that was part of the multi-tier analysis has informed my teaching of the use of qualitative analysis software. This was made explicit at a workshop ('How to Develop an Orthogonal Category System') run for the Association of Qualitative Research in July 1999. This use of dimensions needs further exploration particularly in the context of the use of qualitative data analysis software. Software makes coding on many dimensions possible because it is so easy to code a piece of text to many categories, and this can fundamentally change the way in which categories are used in qualitative analysis. This possibility was raised by Richards & Richards in 1994 but few texts that discuss coding and the use of software in detail (eg Fielding & Lee 1998, Creswell 1998, Richards 1999) highlight the possibilities.

The development of this research method and the associated software in Hypercard has informed the development of NVivo, N4 and N5 through my work as a trainer and consultant in the use of QSR software. The use of multi-tier analysis is fully supported by NVivo as memos are treated as data. The uses of this technique need further investigation and it is already clear that the way in which the position of data is blurred in NVivo opens up the possibilities for new approaches to qualitative analysis.

Both the construction of a chronology and the development of web presentations are promising approaches for other researchers. However, further work needs to be done to investigate how they can be supported through commercially available qualitative data analysis software. Dicks and Mason are working on a Hypermedia Ethnography Project that incorporates hypertext and multimedia (Dicks & Mason 1998, Mason & Dicks 1999). They suggest that hypermedia should be seen as offering an approach encompassing both analysis and presentation but as this is not currently supported by one piece of software, their current focus is on the 'hypertextual' reworking of an existing ethnography using a multimedia authoring tool. There is scope for extending the approach used in this research by converting an analysis produced in a commercial qualitative analysis program into a hypermedia presentation using web technology.

9.2 EVALUATION OF KEY INFLUENCES ON SOFTWARE DEVELOPMENT

Chapter 8 concludes that most software developments will be affected by three dominant influences (individual, organisation and method) that will have a pervasive effect (positive or negative) on the course of the development. Developments will also be critically affected by some of a further ten influences (IT profession, belief, resources, domain, technology, design, vision, change, communication and chance) with large developments affected by many of these influences. This section examines the research that led to this result, discusses the findings in the light of contemporary research and considers the implications for the practice of software development.

9.2.1 Research method

The key influences emerged from a detailed analysis of the software development in X. Data about the software development was collected over seventeen months from taped conversations and system documentation. Transcripts of the conversations and the text of documents were analysed as described in chapter 5. These key influences were compared with the influences emerging from the data collection, the results of a think tank at Object Technology '95 and a summary analysis of the other five longitudinal cases.

The research quality criterion of *goodness* means that the results have to be consistent with the world-views of the participants in software development and credible with anyone involved in software development (specialist developers, users and managers). The *utility* criterion means that it is necessary for the reader to be able to transfer the results of the research to other situations.

As a former member of the Bank of England where I worked as a user of financial information systems and then as a software developer, I was well-placed to understand the world-views of all the participants in the development at X. Although I had not worked

with the specific technology being used by the project, I had used similar tools. The research method was designed to allow the results to be credible with the target audience.

The study of X suffered from the following limitations:

- Access to X was via a manager in GIS and the main contact at X was the project leader Gordan. Although I had conversations with many users this, together with my background, could have favoured the technical view.
- The technique used to identify informants meant that I did not interview any users from Claims or most of the senior managers.
- It was not possible to visit users overseas because of the cost.
- X was selected for detailed analysis because the period studied included the construction and delivery of a complete phase but data was only collected retrospectively about the conception of the overall system.

I was aware of the drawbacks during analysis and although they will have affected the story of the development in X, I do not feel they have affected the *goodness* of the identification of key influences.

The *utility* of the results depends upon the reader being able to apply the results to other software developments. The web presentation of the longitudinal case at X allows the reader to become familiar with the case but this is not enough if the reader's software development area is very different in size or problem domain. Chapter 8 compared the results with the other cases to demonstrate that the results were applicable to other types of software development.

One of the drawbacks with this research was the time taken to collect the data and to analyse the results. Without a longitudinal study it would have been impossible to highlight different perceptions of events when viewed in anticipation, contemporaneously or retrospectively. Without conversations with different participants, the effects of multiple realities would not have been observed. A less detailed analysis, using for example notes of conversations rather than transcripts, would not have revealed some of the less obvious results such as the relatively little talk about the team.

One problem with research that takes a long time is that the world changes during the research. Although technology has changed during the time of this research, the key influences are such that their *utility* is unlikely to be affected by the elapsed time. In discussing the findings the next section will address this issue with reference to current literature.

9.2.2 Implications of the findings for research

The discussion in chapter 8 raised five questions arising from the discussion of the key influences. This section addresses each of the questions in terms of current literature and potential for future research.

a Do developers focus on things they can control at expense of considering other more important influences?

One of the consequences of the positivist paradigm is the desire in research to look for causal relationships. This focuses research on things that developers can do to improve the process. In contrast this research regards the key influences not as things to get right but things to be aware of. In order to compare the thirteen key influences with contemporary literature, three papers were selected that take different perspectives on the problem of project success and failure.

Reel (1999), in the words of the editor of the special issue of IEEE Software on 'Critical success factors in software development' has 'distilled years of experience into a readable essay on critical success factors in software projects' (Bytheway 1999). In doing this Reel draws on ten signs of IS project failure. None of these are explicitly concerned with organisational issues, design or the nature of the IT profession and four are failures in development methods. In his advice, presented as five critical success factors, he recognises the importance of individual skills but still talks about 'building the right team'. Good communication and maintaining the belief of those involved in the development are integral to his recommendations but he does not refer to the organisation or the nature of the IT profession.

Jiang & Klein (2000) present a summary of software development risks distilled from a variety of authors in an attempt to link risk to project effectiveness. The risks they identify are concentrated around characteristics of the domain, technology and the team. The only risk connected with the organisation is 'User attitudes' and surprisingly there are no risks directly related to the development methods (although the short descriptions provided in the paper can be ambiguous). Interestingly, their sample (86 project managers who were members of the Project Management Institute in the US) 'generally found that the risks associated with system development did not impact their more recent project to a great extent' (Jiang & Klein 2000 p6).

Verner, Overmyer & McCain (1999) investigated the 'state-of-the-practice of software development' by having 'structured discussions' with 20 US professionals working in a variety of different organisations on 34 developments (20 regarded as successful). From this they identify seven broad areas for discussion. These areas just group comments together so their categorisation is not the same as mine but the level of detail means that it is possible to compare the two. Much of the discussion centres around method although design, belief (of customers and users), vision, and staffing are also identified as factors. The only aspect of individual that was identified as important was that personal characteristics of the project manager were a problem in over half the failed projects. Similarly organisational issues were only identified as problematic in failed projects. This paper did highlight that some of the problem areas were outside the control of project managers.

These three papers demonstrate that the key influences identified in this research have a broader base than much existing work. This is probably as a result of method. The work by Vernier, Overmyer & McCain (1999) appears to share a qualitative approach but, by only asking software professionals about projects retrospectively, they limit themselves to one reality.

An answer to the question posed at the beginning of this sub-section, is not that developers focus on things they can control, but that research tends to focus on the more tractable influences identified in figure 8.4. This may be because of the world-view of the researcher;

if there was more management research in software development, then the focus might be different.

b Do teams or individuals build software systems?

The importance of considering stakeholders in the project as individuals is very clear in this research but does not appear in published work where team issues often rate highly (see for example papers discussed in 9.2.2a above). Although there is an assumption that software is written by a team, relatively little research has been done on software development teams (section 2.2.3) although Linberg (1999 p6) reports a growing interest in 'diverse and synergistic' teams. Casual observation of developers shows that they largely work alone at a screen. Software developers do collaborate but they come together only for specific activities such as sharing expertise or solving a problem. Understanding the relationship between the software development team and the individual has important management implications and is worthy of more study. There may be parallels with research on the innovation process, which recognises the value of the individual (eg Trott 1993).

c What are the impacts of organisations on software development?

While the impact of IT on the organisation has been widely studied, the impact of the organisation on the software development process is less widely acknowledged. Given the importance of the organisational influences identified in chapter 8 it seems appropriate to consider the applicability of theories and concepts from organisational behaviour literature to the process of software development; despite some question marks over the practical usefulness of some of this literature (Anderson 1998). To illustrate the potential for further work, three areas in the literature (organisational metaphors, emotion and culture) are discussed.

Morgan (1986) showed how metaphorical analysis can be used to help understand organisational behaviour and to manage and design organisations. For example, it might be easier to understand the behaviour of the underwriters in X if their actions were not viewed through a mechanistic lens. Heiskanen (1993) illustrates how assumptions based on an inappropriate metaphor can be a cause of failure in information systems development

which can be addressed by the kind of radical thinking that comes from using an alternative metaphor.

A number of writers have explored the dominance of the mechanistic metaphor in information systems development which 'assumes decision makers have a complete knowledge and control over cause and effect relationships' (Korac-Boisvert & Kouzmin 1995 p62). There are some indications that this metaphor is even more dominant in the development of other types of software (Zhu 2000, Despres 1996) and that a mechanistic concept of information can persist even when the organisation is not viewed mechanistically (Kamm 1995). This mechanistic view implies that with the right techniques the necessary knowledge can be elicited to develop a software system but Jones & Walsham (1992) argue that there are limits on what can and should be known by developers of systems.

Alternative organisational metaphors have been used to provide tools for software development (eg Organizational Scenarios tool developed by Clegg et al 1996) and to justify alternative software development approaches (Jackson 1987). However, Pascoulis (2001) has demonstrated the difficulty of introducing such approaches within software development because of the predominance of hard systems thinkers. Organisational metaphors have also been used to explain the failure of methodologies to provide a panacea in software development (Wastell 1996). He also uses an analysis of organisational culture to explain the failure of method in a particular case. This kind of analysis could help ensure that methodology introduction was more effective.

There seems to be potential for using metaphors to improve the management of change (Cleary & Packard 1992, Neilsen 1999) but there has been limited application of this to the management of software introduction. However, Taylor-Cummings (1998) used a metaphorical analysis as a starting point for understanding the gap between users and information systems professionals.

Despite the potential for applying metaphorical organisational analysis to the software development process, relatively little research has been reported. For example, of over 900 articles in the BIDS¹ database citing Morgan's 1986 seminal work, less than 20 explicitly discuss the impact of organisations on software development. Moreover, the emphasis in most of these articles is on information systems and the influence of the organisation on the product of software development not the process (eg Metcalfe 1997, Brooks 1997, Campbell 1996).

Emotion has emerged as an area of study in organisational behaviour as the dominance of rationality has declined (Domagalski 1999). It has been argued that emotions both 'characterise and inform' organisational process (Fineman 2000 p1). This concept seems appropriate to use because software can invoke very strong feelings in some people (eg Jenny discusses her 'love hate' relationship with the X system and one of the underwriters in X describes the email system as 'a brilliant tool, I love it, really, really good'). There is little research reported on the application of the organisational emotion literature to software development but the following two examples illustrate how it might be used to explain some of the critical influence of the organisation identified in my findings.

The concept of emotional contagion in which individuals are influenced by the emotional state of others (Domagalski 1999) could illuminate the ways in which computer systems are accepted by users and explain the importance of the 'PR' that Jenny stresses in X.

Sandelands & Boudens (2000) emphasise the importance of group relationships in people's feelings towards work and thus the attitude to computer systems may be affected by how well such systems supports those relationships. This may have been a contributing factor in the widespread adoption by the underwriters in X of electronic mail.

Although culture is a subject that has been much studied, there seems to have been relatively little work done on the effects of culture on software development. Dubé & Robey (1999) provide an interesting study of organisational culture within a large US software

¹ Online citation index accessible via Web of Science (<http://wos.mimas.ac.uk>)

development company. Unfortunately they only apply their findings to team organisation and outsourcing so it is not possible to discover what the implications of culture were on the process of software development. Sharp, Robinson & Woodman (2000) argue strongly for the existence of a culture of software engineering (cf 'IT profession' in the key influences) but only refer to a study looking at the effects of this on software quality management systems.

These three examples from the organisational behaviour literature illustrate the potential for further work on applying organisational behaviour theory to understanding the impact of both the developing and using organisation on software development.

d Do methods focus too much on process at the expense of product?

This research has distinguished between two influences: method and design. In this context method encompasses all the project management, process and quality assurance activities that software developers and their managers undertake in the course of development. Design, taken as a noun, is about the internal and external shape of the system. The findings show that while method is something that developers talk about, design may be implicitly regarded as important but is far less visible. Despite this, design can have a critical effect on software development. This difference in emphasis is reflected in the literature and efforts such as the Capability Maturity Model (Paulk et al 1993) focus on improving software development process but pay less regard to product. The research also illustrated how quality assurance that focuses on process can be ineffective. This suggests that a fruitful avenue for research would be to look for quality processes that focus on product. There has been the suggestion (eg Ciborra 1998) that it would be better to view software development as a process of cultivation rather than construction in which processes of interference and support are used to encourage the growth of the software and its use.

e What are the implications of the multiple realities of software development?

A striking example of the different realities is provided in a paper that presents an interesting view on software project failure from a software developer perspective (Linberg 1999). The project failure that is the centre of the paper was 193% over schedule (27 vs. 14 months) and 419% over budget (\$4.224K vs. \$1.008K) although it had met its quality

objective of no post-release software defects (it was a medical instrument). Despite being presented as a project failure, all of the software developers (who each had at least six years experience) regarded it as either the most successful project they had worked on or the second most successful. If, as Farbey, Land & Targett (1999) conclude, multiple stakeholders can have a significant effect on software development, and associated with each of these stakeholders there is a potentially different reality, then it behoves researchers to consult the different stakeholders in undertaking research.

In general the key influences provide an agenda for research, in computer science, information systems and management, on defining the effects and implications of each influence. This research has highlighted the relative paucity of management literature on software development but the social nature of many of the key influences indicates the scope for applying management theories to the specific problems of software development. The next section considers the practical implications of the findings.

9.2.3 Implications for the practice of software development

Implications of the key influences for stakeholders with a significant interest in software development are suggested below. These examples have been drawn from personal experience as a software developer, manager and teacher.

Specialist software developers (software engineers, systems developers)

The key influences provide software developers with a list of influences to consider, not a set of prescriptive actions. The summary in section 8.4.1 suggests ways in which the key influences can be important.

Recognition that the key influences of individual and organisation will affect software development no matter how good the methods, puts effort expended on method into a more appropriate perspective. For example, although users may not know what they want if they do not believe in what they are being offered then no matter how good the system is technically they may not use it (as illustrated by the underwriters in X).

The importance of viewing software building as an individual effort supports modern development techniques of modularisation and incremental development which keep components small enough to be developed by one person. This seemed to be the approach that was being adopted in Z. However, the power of individual developers to make or mar a project means that software developers need to take responsibility for their contribution to the whole system.

Software project managers

The importance of individuals to the development coupled with the general characteristics of IT professionals should influence how each person is managed. It also means that software project managers need to recognise the system's vulnerability to the actions of individuals. The way in which Gordan in X did not know that the system was being completely rewritten, illustrates how hard it can be to manage software professionals. The role of the team in software development can be seen as a device for social contact, not just as an organising method for software development.

Users and managers involved in software development

Most users see software development as a technical activity that they can only engage with in a limited capacity. A recognition that many of the key influences are not technical should help users to participate on a more equal basis. Equally, non-technical managers seem to view software development as a technical activity that they can not manage (Neville's handling of Oscar in Y are a good example of the problems this can cause). The key influences show that managers have much to offer in engaging with the process of software development.

Computer Science education

Computer Science education is still largely technical (cf Quality Assurance Agency benchmark for Computing, 2000) and several of the software engineers with Computing degrees, who I spoke to in X and Y, complained about the limited relevance of their courses. Most of the specialist software developers on the Multinational project in X did not have a Computing qualification but had learnt on the job. However, one barrier to changing the education of computer scientists highlighted in chapter 1, is that computer science students tend primarily to be interested in technical subjects.

The attitude of computer science specialists can be changed using affective learning techniques. For example, drawing on this research I was commissioned to develop a two-week in-house company course to expose software engineers to the more human aspects of software development. The 'Software Assault Course' was an accredited MSc-level module for the University of Hertfordshire. Evaluation three months after the course indicated that the students were much more aware of the wider issues that are important when developing software. More recently, I have been commissioned by a different company to design a course ('Making design visible') to raise the profile of design amongst software engineers (reflecting the relative influences of the key influences 'method' and 'design').

Management education

Managers in X who I spoke to in the course of this project deplored the lack of appropriate education and literature for managers about software development but studies have shown that 'there may be a significant and fundamental gap in management's understanding of software development' (Glass, Vessey & Conger 1992). The key influences suggest a useful framework for educating managers about software development.

9.3 FINAL THOUGHTS

During the course of this thesis the use and acceptability of alternative research paradigms and qualitative approaches to data collection and analysis has increased in both management and information systems (Goles & Hirschheim 2000). Even in computer science there is some movement towards a less positivist approach for research (Fuggetta 1999). However I believe there may be some danger that the reaction to these results will be 'so what?' because 'by making the implicit explicit we articulate something that is perceived to be obviously true, therefore self-evident or commonsense' (Sharp, Robinson & Woodman 2000 p46). I do not believe that '*qualitative* approaches are now accepted as equal in value to quantitative approaches when used appropriately' (Avison et al 1999, original italics). The problem is that while qualitative data has become much more widely used it is often associated with a post-positivist research question.

A move towards a pragmatic approach has been suggested (Goles & Hirschheim 2000) to 'get past the debilitating effects of the positivism versus anti-positivism debate' (Wicks & Freeman 1998 p138). Although I agree that the research approach can, and needs to, be selected to match the research question, I fear that the pragmatic approach may conceal the real differences between qualitative data collected within a post-positivist paradigm and a constructivist paradigm. I collected qualitative data about *W* and *X* but they provide different information because the filter that I was applying when I collected it was different. I have observed a similar difference in a team research project, for which I was a consultant, where the content and interpretation of interviews varied considerably according to the research paradigm of the interviewer.

In discussing this research with other software managers the study of software development as a human activity seems a very natural thing to do. This is not reflected in the concerns of many academic computer scientists who equate social aspects of computing with human-computer interaction (Low et al 1996). Users and non-technical managers are surprised that an activity that they perceive as technical can be viewed as a human activity. Reactions to presentations about my work to management researchers seem to indicate a lack of awareness of its relevance to management. One of the reasons for devising a detailed research method was to give the research credibility to these varied audiences. I believe that there is much to be gained from opening up software development so that it is more widely understood and I trust that this research will contribute to that process.

At the time I started to collect qualitative data it was suggested that PhD students did not do qualitative research because of the time it would take. Indeed I might not, if I had known in advance quite how long the research was going to take, or the radical effect the paradigm shift would have. It is not that I would want to tackle the question in an essentially different way, but I might have asked a different question. In practice the outcomes of this research at a personal level have been mostly positive. Through my work in teaching the use of qualitative software I have become involved in many different research projects around the world. I have also become a better software project manager.

Bibliography

- ABRC (1993) Nature of the PhD: Discussion Document, Advisory Board for the Research Councils
- Anderson,N (1998) 'The people make the paradigm', *Journal of Organizational Behavior*, 19(4) pp323-328
- Antill,L (1985) 'Selection of a Research Method', in: Mumford,E Hirschheim,R Fitzgerald,G & Wood-Harper,T (eds), *Research Methods in Information Systems*, Amsterdam: North-Holland
- Avison,DE & Fitzgerald,G (1988) *Information Systems Development: Methodologies, Techniques and Tools*, Oxford: Blackwell Scientific Publications
- Avison,DE & Myers,MD (1995) 'The Interdisciplinary Nature of Information Systems and the Case for Anthropology', *University of Southampton Discussion Papers in Accounting and Management Science*, Number 95-101
- Avison,DE & Wood-Harper,AT (1990) *Multiview: An exploration in Information Systems Development*, Oxford: Blackwell Scientific Publications
- Avison,D Lau,F Myers,M & Neilsen,PA (1999) 'Action Research', *Communications of the ACM*, 42(1) pp94-97
- Avison,DE Kendall,JE & DeGross,JI (1993) *Human, Organizational and Social Dimensions of Information Systems Development*, Amsterdam: North-Holland
- Backhouse,J & Dhillon,G (1996) 'Structures of responsibility and security of information systems', *European Journal of Information Systems*, 5 pp2-9
- Baets,W (1992) 'Aligning information systems with business strategy', *Journal of Strategic Information Systems*, 1(4) pp205-213
- Baille,EJ (1992) *Towards a Satisfaction Relation between CCS Specifications and their Refinements*, PhD thesis University of Hertfordshire
- Barnes,DJ Buckland,BK & Brancheau,JC (1992) 'Methodological Issues in Emerging Technologies Research: experiences and recommendations', in: *Proceedings of the 25th Hawaii International Conference on System Sciences*, IEEE Computer Society Press, Vol 4 pp325-336

- Belbin,RM (1981) *Management Teams: Why they succeed or fail*, Heinemann
- Belbin,RM (1993a) 'A reply to the Belbin Team-Role Self-Perception Inventory by Furnam, Steele and Pendleton', *Journal of Occupational and Organizational Psychology*, 66 pp259-260
- Belbin,RM (1993b) *Team Roles at Work*, Butterworth-Heinemann
- Bennett,R (1982) *Annotated bibliographic guide to research method and related literature relevant to management studies*, Working Paper No 18 Management Research: A Bibliographic Guide
- Bennett,J & Hinton,P (1994) 'Developing an Information Systems Strategy',in: ISTIP '94, Sunningdale April
- Bennett,J Berenbaum,R Hinton,P Tagg,C & Ward,BK (1996) 'B.CCD: a framework to enable SMEs to make Business-Critical Computing Decisions' presented at BIT'96, November
- Benyon,D & Skidmore,S (1987) 'Towards a Tool Kit for the Systems Analyst', *The Computer Journal*, 30(1) pp2-7
- Beynon-Davies,P (1997) 'Ethnography and information systems development: Ethnography of, for and within IS development', *Information and Software Technology*, 39 pp531-40
- Bloomfield,BP & Vurdubakis,T (1994a) 'Editorial: The Social and the Technical: An Introduction', *Information Technology & People*, 7(1) pp4-8
- Bloomfield,BP & Vurdubakis,T (1994b) 'Boundary Disputes: Negotiating the Boundary between the Technical and the Social in the Development of IT Systems', *Information Technology & People*, 7(1) pp9-24
- Boar,BH (1984) *Application Prototyping*, John Wiley & Sons
- Boland,RJ & Day,WF (1989) 'The experience of systems design: a hermeneutic of organizational action', *Scandinavian Journal of Management*, 5(2) pp87-104
- Brooks,FP Jr (1975) *The Mythical Man-Month, Essays on Software Engineering*, Reading, MA: Addison-Wesley
- Brooks,FP (1987) 'No Silver Bullet', *Computer*, April pp10-19
- Brooks,L (1997) 'Structuration theory and new technology: analysing organizationally situated computer-aided design (CAD)', *Information Systems Journal*, 7 pp133-151
- Brown,J & Tagg,C (1988) 'A Case Study Approach to Teaching Systems Development', in: ISTIP'88 proceedings, Sunningdale March 1988

- Bruner, J (1986) *Actual Minds, Possible Worlds*, Cambridge Massachusetts: Harvard University Press
- Bryman, A (1989) *Research Methods and Organizational Studies*, London: Unwin
- Burgess, C & Hughes, J (1996) 'Human Resource Management Interventions and the New Role of the Systems Analyst', presented at: Open University Business School Human Resource Management Conference, 1-2 April
- Burgoyne, JG (1994) 'Stakeholder Analysis' in: Cassell, C & Symon, G (eds) *Qualitative Methods in Organizational Research: A Practical Guide*, London: Sage, pp187-207
- Burrell, G & Morgan, G (1979) *Sociological Paradigms and Organisational Analysis*, London: Heinemann
- Button, G & Sharrock, W (1994) 'Occasioned practices in the work of software engineers', in: Jirotko, M & Goguen, J (eds) *Requirements Engineering: Social and Technical Issues*, London: Academic Press pp201-216
- Buzan, T (1989) *Use Your Head*, BBC Books
- Bytheway, AJ (1999) 'Successful Software Projects and How to Achieve Them', *IEEE Software*, May/June pp15-17
- Calloway, LJ and Ariav, G (1991) 'Developing and using a Qualitative Methodology to Study Relationships among Designers and Tools', in: Nissen, H-E Klein, HK and Hirschheim, R (eds) *Information Systems Research: Contemporary approaches and emergent traditions*, Amsterdam: North Holland Elsevier, pp175-193
- Campbell, H (1996) 'A Social Interactionalist Perspective on Computer Implementation', *Journal of the American Planning Association*, 62(1) pp99-107
- Cassell, C & Symon, G (1994) 'Qualitative Research in Work Contexts' in: Cassell, C & Symon, G (eds) *Qualitative Methods in Organizational Research: A Practical Guide*, London: Sage, pp1-13
- Chamberlain, K (1995) [k.chamberlain@massey.ac.nz] 'Re: What is grounded theory (longish)', in: QUALRS-L [QUALRS-L@UGA.CC.UGA.EDU], 16 November
- Checkland, P & Scholes, J (1990) *Soft Systems Methodology in Action*, John Wiley & Sons
- Checkland, P (1981) *Systems Thinking, Systems Practice*, John Wiley & Sons
- Checkland, P (1991) 'From Framework through Experience to Learning: the essential nature of Action Research', in: Nissen, H-E Klein, HK and Hirschheim, R (eds) *Information Systems Research: Contemporary approaches and emergent traditions*, Amsterdam: North Holland Elsevier, pp397-403

- Checkland,P (1992) 'Information Systems and Systems Thinking: Time to Unite?', in: Cotterman,WW & Senn,JA (ed) *Challenges and Strategies for Research in Systems Development*, Chichester: John Wiley & Sons, pp353-364
- Ciborra,CU (1998) 'Crisis and foundations: an inquiry into the nature and limits of models and methods in the information systems discipline', *Journal of Strategic Information Systems*, 7 pp5-16
- Cleary,C & Packard,T (1992) 'The Use of Metaphors in Organizational Assessment and Change', *Group & Organization Management*, 17(3) pp229-241
- Clegg,C Coleman,P Hornby,P Maclaren,R Robson,J Carey,N & Symon,G (1996) 'Tools to incorporate some psychological and organizational issues during the development of computer-based systems', *Ergonomics*, 39(3) pp482-511
- Clement,A & Van den Besselaar,P (1993) 'A Retrospective look at PD Projects', *Communications of the ACM*, 36(4) pp29-37
- Coffey,A Holbrook,B & Atkinson,P (1996) 'Qualitative Data Analysis: Representations of a Technology', *Sociological Research Online*, 1(1)
<<http://www.socresonline.org.uk/socresonline/1/1/4.html>>
- Coleman,D Arnold,P Bodoff,S Dollin,C Gilchrist,H Hayes,F Jeremaes,P (1994) *Object-Oriented Development: The Fusion Method*, Prentice-Hall
- Commission on Management Research (1994) *Building Partnerships: Enhancing the Quality of Management Research*, ESRC
- Conger,SA (1992) 'An Innovation Theory of End-User Computing', in: Cotterman,WW & Senn,JA (ed) *Challenges and Strategies for Research in Systems Development*, Chichester: John Wiley & Sons, pp273-294
- Cook,S Mason,PJ Stamper,RK & Tagg,C (1980) 'The LEGOL Project since 1976', *Computers and Law*, 1980(1)
- Cotterman,WW & Senn,JA (eds) (1992) *Challenges and Strategies for Research in Systems Development*, Chichester: John Wiley & Sons
- Cougar,JD (1973) 'Evolution of Business Systems Analysis Techniques', reprinted in: Yourdon,E (ed) (1982) *Writings of the Revolution: Selected Readings on Software Engineering*, New York: Yourdon Press, pp51-84
- Cramer,D (1994) *Introducing Statistics for Social Research*, London: Routledge
- Creswell,JW (1998) 'Qualitative Inquiry and Research Design', London: Sage
- Crinnion,J (1991) *Evolutionary Systems Development: a practical guide to the use of prototyping within a structured systems methodology*, Pitman

- Curtis,B (1984) 'Fifteen Years of Psychology in Software Engineering: Individual Differences and Cognitive Science', reprinted in: DeMarco,T & Lister,T (eds) (1990) *Software State-of-the-art: Selected Papers*, New York: Dorset House Publishing, pp437-453
- Curtis,B Krasner,H & Iscoe,N (1988) 'A Field Study of the Software Design Process for Large Systems', *Communications of the ACM*, 31(11) pp1268-1287
- Curtis,G (1989) *Business Information Systems, Analysis, Design and Practice*, Addison-Wesley
- Davis,GB (1992) 'Systems Analysis and Design', in: Cotterman,WW & Senn,JA (eds) (1992) *Challenges and Strategies for Research in Systems Development*, Chichester: John Wiley & Sons, pp9-21
- Davis,AM (1995) *201 Principles of Software Development*, New York: McGraw-Hill
- De Marco,T & Lister,T (1987) *Peopleware: productive projects and teams*, Dorset House
- De Marco,T (1978) *Structured Analysis and System Specification*, Yourdon inc
- De Marco,T (1982) *Controlling Software Projects*, Yourdon Press
- Demay,T & King,S (1996) 'BPR and OO: origins and common ground', *Computer Bulletin*, 8(1) pp8-9
- Dengate,G Couger,JD & Weber,R (1990) 'Motivational characteristics of Australian information systems personnel', *Australian Computer Journal*, 22(3) pp77-88
- Denzin,NK & Lincoln,YS (1994) 'Introduction: Entering the Field of Qualitative Research', in: Denzin,NK & Lincoln,YS (eds) *Handbook of Qualitative Research*, Thousand Oaks, California: Sage publications, pp1-17
- Despres,CJ-N (1996) 'Information, technology and culture: an ethnography of information technology and modernist business organization', *Technovation*, 16(1) pp1-20
- Dey,I (1993) *Qualitative Data Analysis: A User-Friendly Guide for Social Scientists*, London: Routledge
- di Gregorio,S (1995) 'Qualitative Approaches in Management Research: A view from Britain', presented at: Australian Association for Social Research second conference, La Trobe University, Melbourne, 8-10 December
- Dick,M & Rouse,A (1994) 'A comparison of practitioner and educator experience in object-oriented training', *TOOLS Pacific '94 Technology of Object-Oriented Languages and Systems*
- Dick,M & Rouse,A (1995) 'The Introduction of Object-Orientation in Australian Software Development Organizations: the Applicability of Diffusion of Innovation Theory', in:

Proceedings of the Third European Conference on Information Systems ECIS '95 Vol 2, Athens, Greece, 1-3 June 1995

Dicks,B & Mason,B (1998) 'Hypermedia and Ethnography:Reflections on the Construction of a Research Approach', Sociological Research Online 3(3)
<<http://www.socresonline.org.uk/socresonline/3/3/3.html>>

Domagalski,TA (1999) 'Emotion in organizations: Main currents', Human Relations, 52(6) pp833-852

Donaldson,L (1998) 'The Myth of Paradigm Incommensurability in Management Studies: Comments by an Integrationist', Organization, 5(2) pp267-272

Dubé,L & Robey,D (1999) 'Software stories: three cultural perspectives on the organizational practices of software development', Accounting Management and Information Technologies, 9 pp223-259

Eardley,A Avison,D & Powell,P (1995) 'An Examination of the Links Between 'Vision-based' Business Strategy and Information Technology Developments', in: proceedings BIT'95, Manchester, November

Earl,MJ (1994) 'The new and the old of business process redesign', Journal of Strategic Information Systems, 3(1) pp5-22

Easterby-Smith,M Thorpe,R & Lowe,A (1991) Management Research: An introduction, London: Sage Publications

Edwards,C & Peppard,JW (1994) 'Business process redesign: hype, hope or hypocrisy?', Journal of Information Technology, 9 pp251-266

Eisenhardt,KM (1989) 'Building Theories from Case Study Research', Academy of Management Review, 14(4) pp532-550

Eva,M (1992) SSADM Version 4: A User's Guide, McGraw-Hill

Evans,KM (1968) Planning Small Scale Research, Nfer-Nelson

Farbey,B Land,F & Targett,D (1999) 'Moving IS evaluation forward: learning themes and research issues', Journal of Strategic Information Systems, 8 pp189-207

Farhoomand,AF (1992) 'Scientific Progress of Management Information Systems', Information Systems Research: Issues, methods and practical guidelines, Galliers,R (ed), Blackwell Scientific Publications, Chap. 5 pp93-111

Fayad,ME Hawn,LJ Roberts,MA and Klatt,UR (1993) 'Using the Shlaer-Mellor Object-Oriented Analysis Method', IEEE Software, Vol 10 No 2 pp43-52

Feyerabend,P (1975) Against Method, Verso

- Fidler,C & Rogerson,S (1996) *Strategic Management Support Systems*, London: Pitman Publishing
- Fiedler,KD Grover,V & Teng,JTC (1994) 'Information technology-enabled change: the risks and rewards of business process redesign and automation', *Journal of Information Technology*, 9 pp267-275
- Fielding,NG & Lee,RM (1998) *Computer Analysis and Qualitative Research*, London: Sage
- Fineman,S (ed) (2000) *Emotion in Organizations*, 2nd edn, London: Sage
- Fischer,S Doodeman,M Vinig,T & Achterberg,J (1993) 'Boiling the frog or seducing the fox: organizational aspects of implementing case technology', in: Avison,DE Kendall,JE & DeGross,JI (eds) *Human, Organizational and Social Dimensions of Information Systems Development*, Amsterdam: North-Holland, pp419-435
- Fitzgerald,G Hirschheim,R Mumford,E & Wood-Harper,T (1985) 'Information Systems Research Methodology: An Introduction to the Debate' in: Mumford,EE Hirschheim,R Fitzgerald,G & Wood-Harper,AT (eds), Elsevier Science Publishers
- Flynn,DJ & Goleniewska,E (1993) 'A survey of the use of strategic information systems planning approaches in UK organisations', *Journal of Strategic Information Systems*, 2(4) pp292-319
- Fontana,A & Frey,JH (1994) 'Interviewing: The Art of Science', in: Denzin,NK & Lincoln,YS (eds) *Handbook of Qualitative Research*, Thousand Oaks, California: Sage publications, pp361-376
- Ford,RC & McLaughlin,FS (1992) 'Successful Project Teams: a Study of MIS Managers', *IEEE Transactions on Engineering Management*, 39(4) pp312-317
- Frangos,SA (1998) 'Motivated humans for reliable software products', *Microprocessors and Microsystems* 21 pp605-610
- Frost,P & Stablein,R (eds) (1992) *Doing Exemplary Research*, Sage
- Fuggetta,A (1999) 'Rethinking the modes of software engineering research', *Journal of Systems and Software*, 47 pp133-138
- Furnham,A Steele,H & Pendleton,D (1993a) 'A psychometric assessment of the Belbin Team-Role Self-Perception Inventory', *Journal of Occupational and Organizational Psychology*, 66 pp245-257
- Furnham,A Steele,H & Pendleton,D (1993b) 'A response to Dr Belbin's reply', *Journal of Occupational and Organizational Psychology*, 66 pp261
- Gable,GG (1994) 'Integrating case study and survey research methods: an example in information systems', *European Journal of Information Systems*, 3(2), pp112-126

- Galliers,RD (1992) 'Choosing Information Systems Research Approaches', in: (ed) Galliers,RD Information Systems Research: Issues, methods and practical guidelines, Oxford: Blackwell Scientific Publications, pp144-162
- Galliers,RD (1993a) 'Research Issues in information systems', Journal of Information Technology, 8 pp92-98
- Galliers,RD (1993b) 'Doctoral information systems research in Britain: a report on the UK Information Systems Doctoral Consortiums, 1991-1993', Journal of Information Technology, 8 pp118-120
- Galliers,RD (1995) 'Information Management: Taking It Seriously', in: proceedings BIT'95, Manchester, November
- Gane,C & Sarson,T (1979) Structured Systems Analysis: Tools and Techniques, Prentice-Hall
- Gardiner,PD (1993) 'Researching Conflict in Project Management: A Methodological Case Study' presented at British Academy of Management Conference, Milton Keynes 20-22 September 1993
- Gause,DC & Weinberg,GM (1989) Exploring Requirements: Quality Before Design, New York: Dorset House Publishing
- Gilb,T (1988) Principles of Software Engineering Management, Addison-Wesley
- Gill,J & Johnson,P (1991) Research methods for Managers, Paul Chapman Publishing
- Glass,RL Vessey,I & Conger,SA (1992) 'Software tasks: Intellectual or clerical?', Information & Management, 23 pp183-191
- Goles,T & Hirschheim,R (2000) 'The paradigm is dead, the paradigm is dead ... long live the paradigm: the legacy of Burrell and Morgan', Omega The International Journal of Management Science, 28 pp249-268
- Guba,EG & Lincoln,YS (1994) 'Competing Paradigms in Qualitative Research', in: Denzin,NK & Lincoln,YS (eds) Handbook of Qualitative Research, Thousand Oaks, California: Sage publications, pp105-117
- Gummesson,E (1991) Qualitative Methods in Management Research, Newbury Park, California: Sage Publications
- Hammer,M & Champy,J (1995) Reengineering the Corporation: A Manifesto for Business Revolution, London: Nicholas Brealey
- Hammersley,M & Atkinson,P (1995) Ethnography: principles in practice, 2nd edn, London: Routledge

- Hammond,V (1985) (ed) *Current Research in Management*, London and Dover: Frances Pinter
- Harre,R (1981) 'The positivist-empiricist approach and its alternative', in: Reason,P & Rowan,J (eds) *Human Inquiry: A sourcebook of new paradigm research*, John Wiley & Sons, pp3-17
- Heiskanen,A (1993) 'Organizational Metaphors and Information Systems Practice: A Case Example of Implementation Strategy Formulation', in: Avison,DE Kendall,JE & DeGross,JI (eds) *Human, Organizational and Social Dimensions of Information Systems Development*, Amsterdam: North-Holland, pp399-418
- Henry,SM & Stevens,KT (1999) 'Using Belbin's leadership role to improve team effectiveness: An empirical investigation', *Journal of Systems and Software*, 44 pp241-250
- Heshusius,L (1990) 'Discussion on Criteria', in: Guba,EG (ed) *The Paradigm Dialog*, Newbury Park, California: Sage Publications, pp198-201
- Hirschheim,R & Klein,HK (1989) 'Four Paradigms of Information Systems Development', *Communications of the ACM*, 32(10) pp1199-1216
- Hirschheim,R (1985) 'Information Systems Epistemology: An Historical Perspective', in: Mumford,E Hirschheim,R Fitzgerald,G & Wood-Harper,T (eds) *Research Methods in Information Systems*, Amsterdam: North Holland
- Hoare,CAR (1981) 'The Emperor's Old Clothes', reprinted in: Yourdon,E (ed) (1982) *Writings of the Revolution: Selected Readings on Software Engineering*, New York: Yourdon Press, pp187-199
- Hoare,CAR (1984) 'Programming: Sorcery or Science?', *IEEE Software*, April pp6-16
- Holloway,JA Skinner,DO & Tagg,C (1995) 'Should qualitative research techniques have a place in a manager's toolkit?', working paper presented at: British Academy of Management Conference, Sheffield: 11-13 September
- Hopkins,JB and Spaul,MWJ (1994), 'Critical Thinking in Information Systems Teaching: A Political Agenda?', *Information Systems Teaching Improving the Practice*, Sunningdale, England, March 1994, pp3-7
- Hovenden,FM Yates,S Woodman,M & Sharp,HC (1994) 'The use of ethnography with discourse analysis in the study of software quality management systems', in: Ross,M Brebbia,CA Staples,G & Stapleton,J (eds) *Software Quality Management II Vol 1: Managing Quality Systems*, Southampton: Computational Mechanics Publications
- Hughes,J (1990) *The Philosophy of Social Research*, Longman
- Hughes,JA Somerville,I Bentley,R & Randall,D (1993) 'Designing with Ethnography - Making Work Visible', *Interacting with Computers*, 5(2) pp239-253

- Humphrey,WS (1990) *Managing the Software Process*, Addison-Wesley
- Hyman,RB (1993) 'Creative Chaos in High-Performance Teams: An Experience Report' , *Communications of the ACM*, 36(10) pp 57-60
- Iivari,J (1991) 'A paradigmatic analysis of contemporary schools of IS development', *European Journal of Information Systems*
- Ince,DC (1993) 'RE: research methods in computer science', private email communication 25/11/93
- Ince,D Sharp,H & Woodman,M (1993) *Introduction to Software Project Management and Quality*, London: McGraw-Hill
- Ives,B & Olsen,MH (1981) 'Manager or Technician?: The Nature of the Information Systems Manager's Job', *MIS Quarterly*, 5(4) pp49-63
- Jackson,MA (1992) 'The General and the Particular', in: Cotterman,WW & Senn,JA (eds) (1992) *Challenges and Strategies for Research in Systems Development*, Chichester: John Wiley & Sons, pp33-40
- Jackson,MC (1987) 'Systems Strategies for Information Management in Organizations which are not Machines', *International Journal of Information Management*, 7 pp187-195
- Jacobson,I Booch,G & Rumbaugh,J (1999) *The Unified Software Development Process*, Reading: Addison Wesley Longman
- Janesick,VJ (1994) 'The Dance of Qualitative Research Design: Metaphor, Methodolatry, and Meaning' in: Denzin,NK & Lincoln,YS (eds) *Handbook of Qualitative Research*, Thousand Oaks, California:Sage publications, pp209-219
- Jayaratna,N (1994) *Understanding and Evaluating Methodologies*, Maidenhead, UK: McGraw-Hill
- Jenkins,AM (1985) 'Research Methodologies and MIS Research', in: Mumford,E Hirschheim,R Fitzgerald,G & Wood-Harper,T (eds), *Research Methods in Information Systems*, Amsterdam: North-Holland, pp103-117
- Jiang,J & Klein,G (2000) 'Software development risks to project effectiveness',*Journal of Systems and Software*, 52 pp3-10
- Jirotko,M & Goguen,J (eds) (1994) *Requirements Engineering: Social and Technical Issues*, London: Academic Press
- Jones,M & Walsham,G (1992) 'The Limits of the Knowable: Organizational and Design Knowledge in Systems Development', *IFIP Transactions A: Computer Science and Technology*, 8 pp195-213

- Kaiser,KM & Bostrom,RP (1982) Personality Characteristics of MIS Project Teams: An Empirical Study and Action-Research Design', *MIS Quarterly*, December pp43-60
- Kamm,R (1995) 'Information and the Mechanistic Metaphor: The Place of Information in Organizational Thought', *Systems Practice*, 8(5) pp517-536
- Keen,PGW (1991) 'Relevance and Rigor in Information Systems Research: Improving Quality, Confidence, Cohesion and Impact', in: Nissen,H-E Klein,HK and Hirschheim,R (eds) *Information Systems Research: Contemporary approaches and emergent traditions*, Amsterdam:North Holland Elsevier, pp27-49
- Kendall,JE & Avison,D (1993) 'Emancipatory Research Themes in Information Systems Development: Human, Organizational and Social Aspects', in: Avison,DE Kendall,JE & DeGross,JI (eds) *Human, Organizational and Social Dimensions of Information Systems Development*, Amsterdam: North-Holland, pp1-12
- Kendall,JE & Kendall,KE (1994) 'Metaphors and their meaning for information systems development', *European Journal of Information Systems*, 3(1) pp37-47
- Kirk,J & Miller,ML (1986) *Reliability and Validity in Qualitative Research*, Beverly Hills, California: Sage Publications
- Klein,HK & Welke,RJ (1982) 'Information Systems as a Scientific Discipline', *Proceedings, ASAC*, University of Ottawa, May 1-3, pp106-116
- Kling,R (1996) *Computerization and Controversy: Value Conflicts and Social Choices*, 2nd edn, San Diego: Academic Press
- Kolb,DA (1984) *Experiential Learning: Experience as the Source of Learning and Development*, Prentice-Hall
- Korac-Boisvert,N & Kouzmin,A (1995) 'IT Development: Methodology Overload or Crisis?', *Science Communication*, 17(1) pp57-89
- Kuhn,S & Muller,MJ (1993) 'Participatory Design - Editorial', *Communications of the ACM*, 36(4) pp26-28
- Kvale,S (1996) *InterViews: An introduction to qualitative research interviewing*, Thousand Oaks: Sage
- Land,FF Le Quesne,PN & Wijegunaratne,I (1992) 'Technology Transfer: Organizational Factors Affecting Implementation - Some Preliminary Findings', in: Cotterman,WW & Senn,JA (eds) *Challenges and Strategies for Research in Systems Development*, Chichester: John Wiley & Sons, pp65-81
- Linberg,KR (1999) 'Software developer perceptions about software project failure: a case study', *Journal of Systems and Software*, 49 pp177-192
- Lincoln,YS & Guba,EG (1985) *Naturalistic Inquiry*, Beverly Hills: Sage Publications

Lincoln,YS (1990) 'The Making of a Constructivist: A Remembrance of Transformations Past', in: Guba,EG (ed) *The Paradigm Dialog*, Newbury Park, California: Sage Publications, pp67-87

Little,SE (1991) 'Grounded Theory: Discussant's Comments', in: Nissen,H-E Klein,HK and Hirschheim,R (eds) *Information Systems Research: Contemporary approaches and emergent traditions*, Amsterdam:North Holland Elsevier, pp227-233

Little,SE (1993) 'The Organizational Context of Systems Development', in: Avison,DE Kendall,JE and DeGross,JI (eds) *Human, Organizational and Social Dimensions of Information Systems Development*, Amsterdam:North-Holland, pp439-454

Loomes,M (1991) *Software Engineering Curriculum Design*, PhD thesis, University of Surrey

Low,J & Woolgar,S (1993) 'Managing the Social-Technical Divide: Some aspects of the discursive structure of information systems development' in: Quintas,P (ed) *Social Dimensions of Systems Engineering: people, processes, policies and software development*, Ellis Horwood, pp34-59

Low,J Johnson,J Hall,P Hovenden,F Rachel,J Robinson,H & Woolgar,S (1996) 'Read this and change the way you feel about software engineering', *Information and Software Technology*, 38 pp77-87

Lowson,B and Rich,N (1995) 'Reflective approaches to management and organisational research', working paper presented at: British Academy of Management Conference, Sheffield: 11-13 September

Loy,P (1993) 'The Method Won't Save You - but it can help', *Software Engineering Notes*, 18(1) pp30-33

Marshall,C & Rossman,GB (1995) *Designing Qualitative Research*, 2nd edn, Thousand Oaks, California: Sage Publications

Marshall,C (1990) 'Goodness Criteria: Are They Objective or Judgment Calls?', in: Guba,EG (ed) *The Paradigm Dialog*, Newbury Park, California: Sage Publications, pp188-197

Mason,B & Dicks,B (1999) 'The Digital Ethnographer', *Cybersociology*, 6
<<http://www.socio.demon.co.uk/magazine/6/dicksmason.html>>

Massey,A (1996) 'Using the literature: 3 x 4 analogies', *The Qualitative Report*, 2(4)

McCracken,DD & Jackson,MA (1982) 'Life Cycle Concept Considered Harmful' in *ACM Sigsoft Software Engineering Notes*, 7(2) pp 29-32

Meloy,JM (1994) *Writing the Qualitative Dissertation: Understanding by Doing*, Lawrence Erlbaum Associates

- Metcalfe,B (1997) 'Project management system design: A social and organisational analysis', *International Journal of Production Economics*, 52 pp305-316
- Miles,MB & Huberman,AM (1994) *Qualitative Data Analysis*, 2nd edn, Thousand Oaks, California: Sage Publications
- Mintzberg,H (1994) 'The Fall and Rise of Strategic Planning', *Harvard Business Review*, 72(1), pp107-114
- Mishler,EG (1986) *Research Interviewing: Context and Narrative*, Harvard University Press
- Moreton,R (1995) 'Transforming the organization: the contribution of the information systems function', *Journal of Strategic Information Systems*, 4(2) pp149-163
- Morgan,G (ed) (1983) *Beyond Method: strategies for Social research*, Sage
- Morgan,G (1986) *Images of Organization*, Sage
- Morgan,G (1993) *Imaginization: the art of creative management*, Newbury Park: Sage
- Morison,M & Moir,J (1998) 'The role of the computer software in the analysis of qualitative data: efficient clerk, research assistant or Trojan horse?', *Journal of Advanced Nursing*, 28(1) pp106-116
- Morse,JM (1994) 'Designing Funded Qualitative Research', in: Denzin,NK & Lincoln,YS (eds) *Handbook of Qualitative Research*, Thousand Oaks, California: Sage publications, pp220-235
- Mumford,E (1979) *Designing Participatively*, Manchester Business School
- Murray,F & Willmott,H (1993) 'The Communications Problem in Information Systems Development: Towards a relational approach' in: Quintas,P (ed) *Social Dimensions of Systems Engineering: people, processes, policies and software development*, Ellis Horwood, pp165-178
- Murray,F & Woolgar,S (1991) 'Social Perspectives on Software: a preliminary report', CRICT Discussion Paper, Brunel University
- Nash,C (ed) (1990) *Narrative in Culture: The uses of storytelling in the sciences, philosophy, and literature*, London and New York:Routledge
- Naur,P (1971) 'Problem Formulation - the Fertile Soil of the Software Development Project' reprinted in: Naur,P (1992) *Computing: A Human Activity*, New York: Addison-Wesley
- Naur,P (1992) *Computing: A Human Activity*, ACM Press/Addison Wesley
- Niederman,F Brancheau,JC & Wetherbe,JC (1991) 'Information Systems Management Issues for the 1990s', *MIS Quarterly*, 15(4) pp475-495

- Nielsen,S (1999) 'Talking About Change: An Analysis of Participative Discourse Amongst IT Operations Personnel', *Australian Computer Journal*, 31(4) pp102-108
- O'Connell, F (1994) *How to Run Successful Projects*, London: Prentice Hall
- Olson,D (1993) *Exploiting Chaos: Cashing in on the Realities of Software Development*, New York: Van Nostrand Reinhold
- Open University (1998) 'U500 Doing Academic Research', Milton Keynes: The Open University
- Opie,A (1997) 'Teams as Author: Narrative and Knowledge Creation in case discussions in Multi-Disciplinary Health Teams', *Sociological Research Online* 2(3), <<http://www.socresonline.org.uk/socresonline/2/3/5.html>>
- Oppenheim,AN (1966) *Questionnaire Design and Attitude Measurement*, Aldershot, England: Gower
- Oram,I (1998) *Computer Support of Collaborative Case Based Learning by MBA Students*, IET Thesis, Milton Keynes:The Open University
- Orlikowski,WJ & Baroudi,JJ (1991) 'Studying Information Technology in Organizations: Research Approaches and Assumptions', *Information Systems Research*, 2(1) pp1-28
- Orlikowski,WJ & Robey,D (1991) 'Information Technology and the Structuring of Organizations', *Information Systems Research*, 2 pp143-169
- Ould,MA (1990) *Strategies for Software Engineering: The Management of Risk and Quality*, John Wiley & Sons
- Parnas,DL & Clements,PC (1985) 'A rational design process: How and why to fake it', reprinted in: De Marco,T & Lister,T (eds) (1990) *Software State-of-the-art: Selected Papers*, New York: Dorset House Publishing, pp345-357
- Pascoulis,C (2001) *Understanding the Successes, Failures, and Limitations of Adopting SSM in a Software Development Environment*, proposed PhD thesis, University of Hertfordshire
- Paulk,MC Curtis,B Chrissis,MB & Weber,CV (1993) 'Capability Maturity Model for Software, version 1.1', *IEEE Software*, July pp18-27
- Peters,T (1992) *Liberation Management*, Macmillian
- Phillips,DC (1990) 'Postpositivistic Science: Myths and Realities', in: Guba,EG (ed) *The Paradigm Dialog*, Newbury Park, California: Sage Publications, pp31-45
- Phillips,D (1992) *The Social Scientist's Bestiary: A guide to fabled threats to, and defences of, naturalistic social science*, Oxford:Pergamon Press

Poland,BD (1995) 'Transcription Quality as an Aspect of rigor in Qualitative Research', *Qualitative Inquiry*, 1(3) pp290-310

Popper,KR (1970) 'Normal Science and its Dangers', *Criticism and the Growth of Knowledge*, Lakatos,I & Musgrave,A (eds), Cambridge University Press

Preece,J & Rombach,HD (1993) *Human-Computer Interaction (HCI) Measurement Approaches: Towards a Common Framework*, OU Computing Department Technical Report 93/05

Preece,J Rogers,Y Sharp,H Benyon,D Holland,S & Carey,T (1994) *Human-Computer Interaction*, Wokingham, England: Addison-Wesley Publishing Company

Qualtiy Assurance Agency Benchmark for Computing (2000), <<http://www.qaa.ac.uk>>

Quintas,P (1994) 'Programmed Innovation? Trajectories of change in software development', *Information Technology and People*, 7(1) pp25-47

Quintas,P (1996) 'Chapter 3: Software by Design', in: Mansell,R & Silverstone,R (eds) *Communication by Design*, Oxford University Press

Randall,D Hughes,J & Shapiro,D (1993) 'Systems Development - The Fourth Dimension: Perspectives on the social organization of work' in: Quintas,P (ed) *Social Dimensions of Systems Engineering: people, processes, policies and software development*, Ellis Horwood, pp197-214

Reel,JS (1999) 'Critical Success Factors in Software Projects',*IEEE Software*, May/June pp18-23

Richards,L & Richards,T (1994) 'Using Hierarchical Categories in Qualitative Data Analysis', in: Kelle,U (ed) *Computers and Qualitative Methodology*, Sage

Richards,L (1999) *Using NVivo in Qualitative Research*, Bundoora Victoria: Qualitative Solutions and Research

Riessman,CK (1990) *Divorce Talk: Women and Men Make Sense of Personal Relationships*, Rutgers University Press

Robson,W (1994) *Strategic Management and Information Systems*, Pitman

Ross,M & Staples,G (1994) 'Computing Research', in: *Information Systems Teaching Improving the Practice*, Sunningdale, England, March 1994, pp123-125

Rouse,A & Dick,M (1994) 'The Use of Computerised Tools in Qualitative Information Studies', in: Shanks,G & Arnott,D (eds) *Proceedings of Fifth Australasian Conference on Information Systems*, Monash University, Melbourne, Australia

Sandelands,LE & Boudens,CJ (2000) 'Feeling at Work', in: Fineman,S (ed) *Emotion in Organizations*, 2nd edn, London: Sage pp46-63

Sauer,C (1993a) *Why Information Systems Fail: a case study approach*, Henley-on-Thames: Alfred Waller Ltd

Sauer,C (1993b) 'Partial Abandonment as a Strategy for Avoiding Failure', in: Avison,DE Kendall,JE & DeGross,JI (eds) *Human, Organizational and Social Dimensions of Information Systems Development*, Amsterdam: North-Holland, pp143-167

Scholefield,DJ (1992) *A Refinement Calculus for Real-Time Systems*, PhD Thesis University of York

Schwandt,TA (1994) 'Constructivist, Interpretivist Approaches to Human Inquiry', in: Denzin,NK & Lincoln,YS (eds) *Handbook of Qualitative Research*, Thousand Oaks, California: Sage publications, pp118-137

Seddon,PB (1995) *A Re-Specification of the DeLone and McLean Model of IS Success*, Research Paper 95-07, Department of Accounting and Finance, Melbourne University, Australia

Shank, G. (1995) [gshank@niu.edu] 'what is postivism, anyway?', in: QUALRS-L [QUALRS-L@UGA.CC.UGA.EDU], 18 November

Sharp,H Robinson,H & Woodman,M (2000) 'Software Engineering: Community and Culture', *IEEE Software*, January/February pp40-47

Shlaer,S & Mellor,SJ (1988) *Object-Oriented Systems Analysis: Modeling the World in Data*, Yourdon Press

Shlaer,S & Mellor,SJ (1992) *Object Lifecycles: Modeling the World in States*, Yourdon

Siegel,S & Castellan,NJ Jr (1988) *Nonparametric statistics for the behavioural sciences*, 2nd edn, McGraw-Hill

Silverman,D (1993) *Interpreting Qualitative Data: Methods for Analysing Talk, Text and Interaction*, London: Sage

Skinner,D, Tagg,C & Holloway,J (2000) 'Managers and Research: The Pros and Cons of Qualitative Approaches', *Management Learning*, 31(2) pp163-179

Slack,RS (1998) 'On the Potentailities and Problems of a WWW Based Naturalistic Sociology', *Sociological Research Online* 3(2),
<<http://www.socresonline.org.uk/socresonline/3/2/3.html>>

Smith,JB & Weiss,SF (1988) 'Hypertext', *Communications of the ACM*, 31(7) pp816-819

Smith,JK (1990) 'Alternative Research Paradigms and the Problem of Criteria', in: Guba,EG (ed) *The Paradigm Dialog*, Newbury Park, California: Sage Publications, pp167-187

- Sommerville,I (1989) *Software Engineering*, 3rd edn, Wokingham, England: Addison-Wesley Publishing Company
- Sommerville,I (1992) *Software Engineering*, 4th edn, Wokingham, England: Addison-Wesley Publishing Company
- Sparkes,J (1981) 'What is this thing called Science?', *New Scientist*, 15 January, pp156-158
- Spradley,JP & McCurdy,DW (1972) *The Cultural Experience: Ethnography in Complex Society*, Chicago: Science Research Associates
- Stacey,R (1992) *Managing chaos: Dynamic Business Strategies in an Unpredictable World*, Kogan Page
- Stake,RE (1995) 'The Art of Case Study Research', London: Sage
- Stanley,T (1992) *Hypertalk and Hypertext*, Oxford: Butterworth-Heinemann
- Star,SL (1995) *The Cultures of Computing*, Blackwell
- Stevens,P & Pooley,R (1999) *Using UML: Software engineering with objects and components*, Harlow: Addison-Wesley
- Strauss,A & Corbin,J (1990) *Basics of Qualitative Research: grounded theory procedures and techniques*, California: Sage
- Strauss,A & Corbin,J (1994) 'Grounded Theory Methodology: An Overview', in: Denzin,NK & Lincoln,YS(eds) *Handbook of Qualitative Research*, Thousand Oaks, California: Sage publications, pp273-285
- Svantessan,I (1989) *Mind Mapping and Memory*, Kogan Page
- Swanson,EB & Beath,CM (1988) 'The Use of Case Study Data in Software Management Research', *The Journal of Systems and Software*, 8 pp63-71
- Tagg,C (1979) *LEGOL*, Hatfield Polytechnic MSc Dissertation
- Tagg,C (1991) *TBSD Methodolgy*, University of Hertfordshire Technical Report
- Tagg,C (1993) 'A Hypercard Tool for Academic Writing', presented at British Academy of Management Conference, Milton Keynes, September
- Tagg,C (1994a) 'Process Quality Management: A case study of its use in determining the IT strategy for a small business' in: Ross,M Brebbia,CA Staples,G and Stapleton,J (eds) *Software Quality Management II Vol 1: Managing Quality Systems*, Southampton:Computational Mechanics Publications
- Tagg,C (1994b) 'IS/IT needs strategic thinking', in: proceedings BIT'94, Manchester, November

- Tagg,C (1996a) 'Exploring the Qualitative Quagmire for Information Systems Research' presented (and commended as most innovative research paper) at first UKAIS conference, Cranfield, April
- Tagg,C (1996b) 'Interplay between Model and Method', presented at: Essex '96 Fourth International Social Science Methodology Conference, Colchester 1-5 July
- Taylor-Cummings,A (1998) 'Bridging the user-IS gap: a study of major information systems projects', *Journal of Information Technology*, 13 pp29-54
- Tesch,R (1990) *Qualitative Research : Analysis Types and Software Tools*, Basingstoke: Falmer Press
- Thamhain,HJ & Wilemon,DL (1987) 'Building High Performing Engineering Project Teams', *IEEE Transactions on Engineering Management*, EM-34(3) pp130-137
- Tranfield,D & Starkey,K (1998) 'The Nature , Social Organization and Promotion of Management Research: Towards Policy', *British Journal of Management*, 9 pp341-353
- Trott, P. (1993) *Inward Technology Transfer as an Interactive Process: a case study of ICI*, Cranfield University: Thesis
- Tull,DS & Hawkins,DI (1980) *Marketing Research, Measurement and Method*, New York: Macmillan
- UKAIS (1996) *UK Academy for Information Systems Newsletter*, edited by: Avison,D, 2(1)
- Valdez,MEP (1988) *A Gift from Pandora's Box: The Software Crisis*, PhD Thesis, University of Edinburgh
- Van Maanen,J Dabbs,JM & Faulkner,RR (1982) *Varieties of Qualitative Research*, Beverly Hills, California: Sage Publications
- Verner,JM Overmyer,SP & McCain,KW (1999) 'In the 25 years since The Mythical Man-Month what have we learned about project management?', *Information and Software Technology*, 41 pp1021-1026
- Vitalari,NP. (1985) 'The Need for Longitudinal Designs in the Study of Computing Environments' in: Mumford,E Hirschheim,R Fitzgerald,G & Wood-Harper,T (eds), *Research Methods in Information Systems*, Amsterdam: North-Holland, pp243-263
- Walsham G (1991) 'Organizational metaphors and information systems research', *European Journal of Information Systems*,2(1), pp83-94
- Walsham,G (1993) *Interpreting Information Systems in Organisations*, John Wiley & Sons
- Ward,BK (1990) 'Planning for Profit' in Lincoln,T (ed) *Managing Information Systems for Profit*, John Wiley & Sons pp103-146

Ward,J Griffiths,P & Whitmore,P (1990) Strategic Planning for Information Systems, John Wiley & Sons

Wastell,DG (1996) 'The fetish of technique: methodology as a social defence', Information Systems Journal, 6 pp25-40

Weerakkody,J Bennett,J & Tagg,C (1996) 'Key Issues for Successful Exploitation of BP&ISR in Sri Lanka', in: CSSL 15th National Computer Conference proceedings, Colombo 13-15 September

Weerakkody,V Tagg,C & Bennett,J (1995) 'Bridging the Gap Between Business Process Reengineering and Information Systems: Identifying Key Issues', in: proceedings BIT'95, Manchester, November

Wegner,P (1976) 'Research Paradigms in Computer Science', 2nd International Conference on Software Engineering, San Oct. 1976

Weide,BW & Defazio,S (1993) 'A Framework for Modeling Software Engineering Processes', International Journal of Software Engineering and Knowledge Engineering, 3(3) pp351-368

Weinberg,GM (1971) Psychology of Computer Programming, New York: Van Nostrand Reinhold Company

Weinberg,GM (1982) 'Overstructured Management of Software Engineering', reprinted in: DeMarco,T & Lister,T (eds) (1990) Software State-of-the-art: Selected Papers, New York: Dorset House Publishing, pp3-12

Weitzman,EA and Miles,MB (1995) Computer Programs for Qualitative Data Analysis: A software sourcebook, Thousand Oaks California: Sage

Whitaker,K (1994) Managing Software Maniacs: finding, managing, and rewarding a wining development team, New York: John Wiley and Sons

Whitley,R (1984) 'The Scientific Status of Management Research as a Practically Oriented Social Science', Journal of Management Studies, 21(4) pp369-390

Wicks,AC & Freeman,RE (1998) 'Organization Studies and the New Pragmatism: Positivism, ant-positivism, and the Search for Ethics', Organization Science, 9(2) pp123-140

Willcocks,L & Smith,G (1995) 'IT-enabled business process reengineering: organizational and human resource dimensions', Journal of Strategic Information Systems, 4(3) pp279-301

Wood-Harper,T (1985) 'Research Methods in Information Systems: Using Action Research', in: Mumford,E Hirschheim,R Fitzgerald,G & Wood-Harper,T (eds), Research Methods in Information Systems, Amsterdam: North-Holland, pp169-191

Wood-Harper,AT Miles,RK & Booth,PA (1993) 'Designing Research Education in Information Systems: Towards a Global View', in: Khosrowpour,M & Loch,KD (eds) Global

Information Technology Education - Issues and Trends, Harrisburg: Idea Group Publishing, pp453-487

Work,B (1994) 'An Emerging Paradigm for Information Systems?', in: Information Systems Teaching Improving the Practice, Sunningdale, England, March 1994, pp9-13

Yin, RK (1994) Case study Research: Design and Methods, 2nd edn, Thousand Oaks California: Sage

Yourdon,E (1989) Modern structured Analysis, Prentice-Hall

Zachary,WB & Krone,RM (1984) 'Managing Creative Individuals in High-Technology Research Projects', IEEE Transactions on Engineering Management, EM31(1) pp37-40

Zave,P (1991) 'An Insider's Evaluation of PAISLey', IEEE Transactions on Software Engineering, 17(3) pp212-225

Zhu,Z (2000) 'WSR: A Systems Approach for Information Systems Development', Systems Research and Behavioral Science, 17 pp183-203

Glossary of terms used in cases

ATM

Asynchronous Transfer Mode – a protocol for high speed transmission and switching of packet data with the ability to provide assured quality of service, eg deliver an uninterrupted video stream.

broker

Someone who acts on behalf of a client to find them an insurance company.

Dbase

A PC database often used for end-user developments.

CASE

Software to assist software development, usually consisting of a set of interacting drawing tools.

client-server

A systems architecture where the processing of a transaction is split between two or more computers. The client computer, handling presentational and foreground processing, is linked by a network to a server computer handling background and database tasks.

Controlling office

The office in X which underwrote the insurance programme, not the offices that issued individual policies on that programme.

CRUD

A technique for analysing where data is created, read, updated or deleted so that functions which all access the same data can be grouped together.

CUPID

Software developed for V.

entity model

Technique for analysing data, often used as prelude to designing a relational database.

evolutionary prototyping

A technique for developing software using prototypes, where the prototype gradually evolves into the final system. This can be contrasted with throwaway prototyping where a prototype is developed to investigate some aspect of a system (such as the user interface) but is not used in the final system.

frame relay

A high speed data communication technology.

GIS

Group Information Systems, the department in X developing the Multinational system.

insurance programme

A number of insurance policies which together provide comprehensive cover for a multinational company in different countries.

ISDN

Integrated Services Digital Network – a digital telephone network capable of carrying a number of services such as data not just voice.

LAN

Local area network.

Marine

A group in X responsible for marine insurance that was also having a system developed for them by the Multinational team.

MIS

Management Information System.

Multinational Operations

The department of underwriters in X responsible for writing multinational insurance programmes.

Multinational system

System developed in X to handle large multinational insurance programmes.

Novell

PC local area networking product.

Oracle

Professional relational database system that can be used for complete development or as a server in a client server application.

Powerbuilder

A PC based tool for designing screens and often used for building the client part of client-server applications.

PRINCE

Projects IN a Controlled Environment – a structured project management method.

Q&A

A commercial PC based end-user database development tool.

Q+E

A commercial end-user reporting tool for use with Sybase databases.

Re-insurance

A method of spreading the risk of large insurance policies by reinsuring part of the risk with another company.

Replication server

Sybase software which allowed data to be replicated automatically in distributed databases.

router

A device (hardware or software) for directing network traffic along a particular route.

Services

Department in X responsible for servicing insurance programmes by undertaking all the necessary back office type activities to ensure that policies are issued.

Speedy

Existing system in V, so-called because it suffered from performance problems.

structured analysis, structured approach

Use of structured tools such as dataflow diagrams and entity models often linked in a methodology such as SSADM which prescribes how the models should relate to one another.

Sun Accounts

A company offering a range of accounting packages based on using Sybase databases.

switch

A piece of hardware in a network which determines where messages should be sent.

Sybase

Professional relational database system used as a server in the X development.

terminal emulator

A piece of software running on a PC which allows it to behave as a mainframe terminal.

WAN

Wide Area Network.

warm standby

A backup machine that is rapidly available to replace a live machine in case of failure.

A Questionnaire

The following four pages contain the questionnaire on the factors affecting software development discussed in chapter 2. This is the final version of the questionnaire that was sent to students on the CCI course. There were small clarifications in the wording between this and the first version given to the MBA students.

Centre for Information and Innovation

Questionnaire on the factors affecting software development

Clare Tagg, September 1994

This questionnaire is part of a research programme, supervised by Professor Roland Kaye and Dr Jacky Holloway, looking at systems within organisations which include some element of computerisation. The research is examining the interface between management and computer specialists with the aim of improving the development of software systems. The questionnaire identifies the factors which are considered to be important by the people involved in the development of systems that include computers. Analysis of the results will show whether the factors which are considered critical vary according to the organisation, the nature of the system, the level of technical computing knowledge or gender. Please help the research by completing the questionnaire and returning it in the reply paid envelope. All results will be treated in confidence.

Your experience

Please tick the box or boxes which reflect your involvement with the development of systems that include computers:

		Comments
Your role in the development (tick as many boxes as apply)	<input type="checkbox"/> Responsible for the whole system	
	<input type="checkbox"/> Responsible for the computer system	
	<input type="checkbox"/> User of the system	
	<input type="checkbox"/> Computer specialist	
	<input type="checkbox"/> Other, please specify _____	

Your involvement in the development (tick as many boxes as apply)	<input type="checkbox"/> Defining what is required of the system
	<input type="checkbox"/> Setting up non-computerised procedures
	<input type="checkbox"/> Implementing the computer programs
	<input type="checkbox"/> Testing the system
	<input type="checkbox"/> Using the system
	<input type="checkbox"/> Managing the development
	<input type="checkbox"/> Other, please specify _____

Your gender	Male	<input type="checkbox"/>	Female	<input type="checkbox"/>
-------------	------	--------------------------	--------	--------------------------

Your organisation

Please tick the box or boxes which best describe your organisation:

				Comments
Purpose of your organisation (tick as many boxes as apply)	Manufacturing	<input type="checkbox"/>	Service	<input type="checkbox"/>
	Retailing	<input type="checkbox"/>	Local government	<input type="checkbox"/>
	Financial	<input type="checkbox"/>	Central government	<input type="checkbox"/>
	Educational	<input type="checkbox"/>		
	Other, please specify _____			

Size of your organisation	Number of employees < 20	<input type="checkbox"/>
	Number of employees 20 - 99	<input type="checkbox"/>
	Number of employees 100 - 1000	<input type="checkbox"/>
	Number of employees > 1000	<input type="checkbox"/>

Types of computerisation in your organisation (tick as many boxes as apply)	None	<input type="checkbox"/>
	Mainframe or mini computer(s)	<input type="checkbox"/>
	Personal computers (PCs or Macs)	<input type="checkbox"/>
	Some people use a computer in their work	<input type="checkbox"/>
	Most people use a computer in their work	<input type="checkbox"/>
	Electronic mail used by many people	<input type="checkbox"/>
	Use of computers varies	<input type="checkbox"/>

Factors which affect systems development

Please give details of the most recent system, which includes a computer, where you have been involved in the development or enhancement of the system. (Your involvement could have been as user or developer.) If you have had no involvement, please tick here ☐ and go to the end of the questionnaire.

What is the main purpose of the system? _____

What is the main computer hardware and software used? _____

How long did the development take? _____

How many people were involved full time? _____

For this system, please indicate how much the following factors affected the success of the development. If the effect was particularly positive or negative, indicate this by entering + or - in the box.

Factors	No effect	Minor effect	Major effect	Critical effect	No opinion	Comments
Technical competence of the computer specialists	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Involvement of users of the system in the design	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Commitment of computer specialists to the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Commitment of users to the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Commitment of managers to the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Computer resources available during development	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Methods used by computer specialists	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Widespread consultation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Amount of democracy in organisation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Amount of realistic planning used in development	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Consideration of users' jobs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Way in which development team is organised	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Financial health of the organisation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Monitoring and control of development	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Experience of the developers	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Size of development team	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Experience in the organisation of this kind of development	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Factors	No effect	Minor effect	Major effect	Critical effect	No opinion	Comments
Computer equipment (hardware)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Computer programs (software)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Design of the computer screens and reports (user interface)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Testing	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Introduction of the system	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Establishing requirements	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Organisational politics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Factors external to the organisation	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Training	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Organisational change (now or later)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Other, please specify						
_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
_____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Results of the research

Please give your name and address if you would like to receive the results of this research.

Thank you for taking time to complete this questionnaire. Please send any further comments about the research to:

Clare Tagg

Open Business School, Open University, Walton Hall, Milton Keynes, MK7 6AA

tel: 0908 654710, email: c.tagg@open.ac.uk

B Think tank participants

The following people contributed to the think tank at Object Technology '95 on Critical Success Factors in Object Oriented Developments (see 4.3). Their views did not reflect their companies but rather their experience as software developers, educators and technical managers:

Kieran Boyce, AMS Management Systems
David Briant, Objective
Carol Byde, Bass Taverns
Ian Cheyne, Siemens Plessey Systems
Dave Cleal, Object Designers
John Daniels, Object Designers
Peter Doran, Unipart
Avis France, Cincom
Jackie Gombault, Save & Prosper
Steve Hand, University of Portsmouth
Anthony Heritage, BT
Gerald Kreissig, IBM, Germany
Tony Law, SmithKline Beecham
David Lewis, Unipart
Peter Long, Ernst & Young
Tim Mackinnon
Peter Marks, A.S.E. Consultancy
Ian Mitchell, IBM
Colin Moon, Xyratex
Ian Munton, Barclays
Ron Newsham, GPT
Mike Notman, Bass Taverns
Paul Prendergast, NatWest Home Loans
John Ridgway, Serco Project Engineering
Charles Sinclair, ASTA Development Corporation
Jari Turunen, Nokia Mobile Phones, Finland
Mike Welham, Serco Project Engineering

plus 2 others who wish to remain anonymous.

C Qualitative analysis software

This appendix describes the qualitative analysis software that was written in Hypercard to support the research. Hypercard consists of 'cards' collected into stacks. Three stacks were used:

- analysis stack for the construction of the chronology and analysis by direct influence
- synthesis stack for the synthesis memos, construction of the commentary and identification of key influences
- pseudonym stack for the identification of the pseudonyms used in the other two stacks

The relationship between these three stacks is shown in figure C.1. The following sections explain the use of the cards in each of the stacks. The functionality was provided using routines written in Hypertalk, the event-driven programming language built into Hypercard.

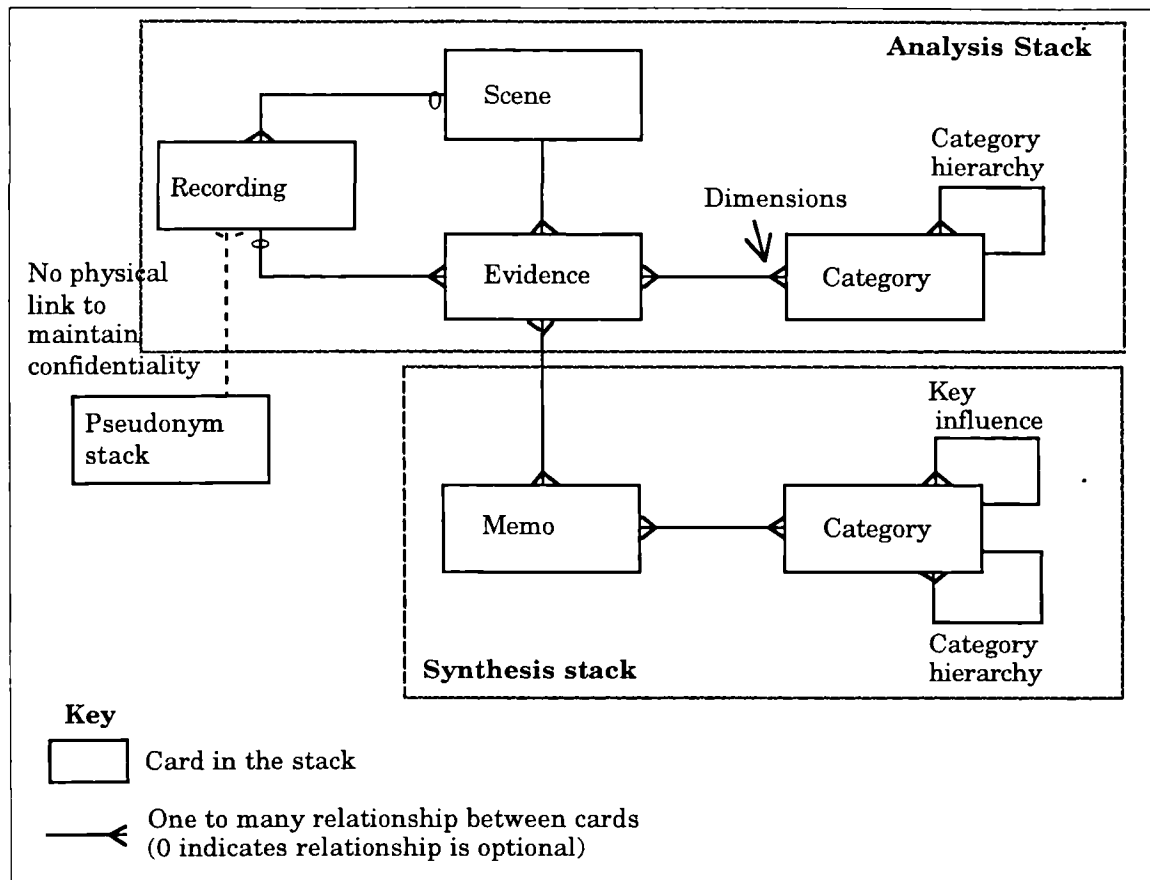


Figure C.1 Relationship between the cards in the Hypercard stacks

C.1 ANALYSIS STACK

The analysis stack was used for:

- recording the details of visits and data collection on scene and recording cards
- entering the transcripts of conversations and case documents on evidence cards
- constructing the chronology by ordering the evidence cards and trimming the content
- categorising the evidence according to the direct social and technical influences using category cards

summarising and exporting to Word the chronology for chapter 6

constructing and exporting scenes, recordings, story, evidence, social and technical influences to the web-site

The functionality in the stack was primarily provided through buttons on each card. Some general functions, such as being able to switch between card types, were provided by menus. The following sub-sections illustrate the main cards and summarise the functionality provided.

C.1.1 Scenes

Figure C.2 illustrates the scene card for an all day visit to the North of England on 14 October 1994. This figure illustrates many of features common to all cards. The buttons on the right hand side allow the user to:

- create new scene cards copying relevant information from the current card
- delete the current scene and associated evidence
- find a scene either based on the content of one of the fields or searching for a word
- mark scenes either based on the content of one of the fields or searching for a word; alternatively scenes may be individually marked by clicking on the marking corner
- sort scenes in a number of orders
- export all or marked scenes in sorted order using rich text format to preserve styles in Word; the styles used are stored on a special preference card
- create a new recordings associated with this scene
- create new evidence associated with the scene as a whole rather than with a recording
- link a scene to an existing recording

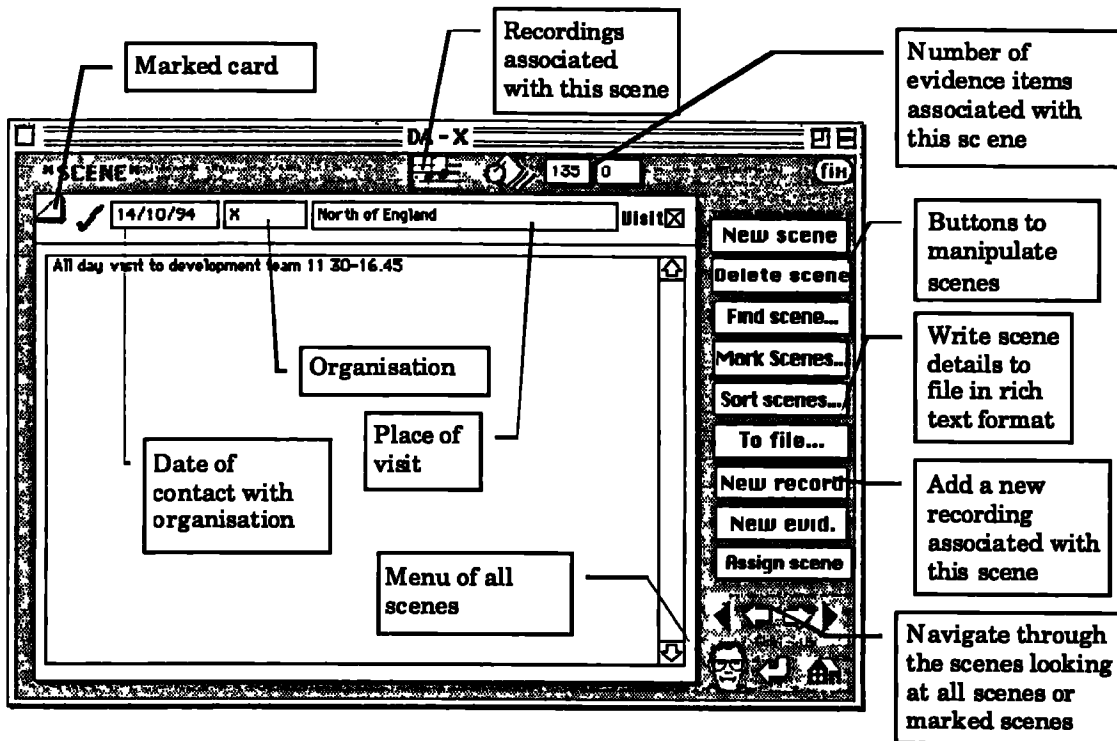


Figure C.2 Scene card from the analysis stack

C.1.2 Recordings

Associated with each scene there are recording cards for each separate conversation or case document. Figure C.3 shows one recording from the scene shown in figure C.2.

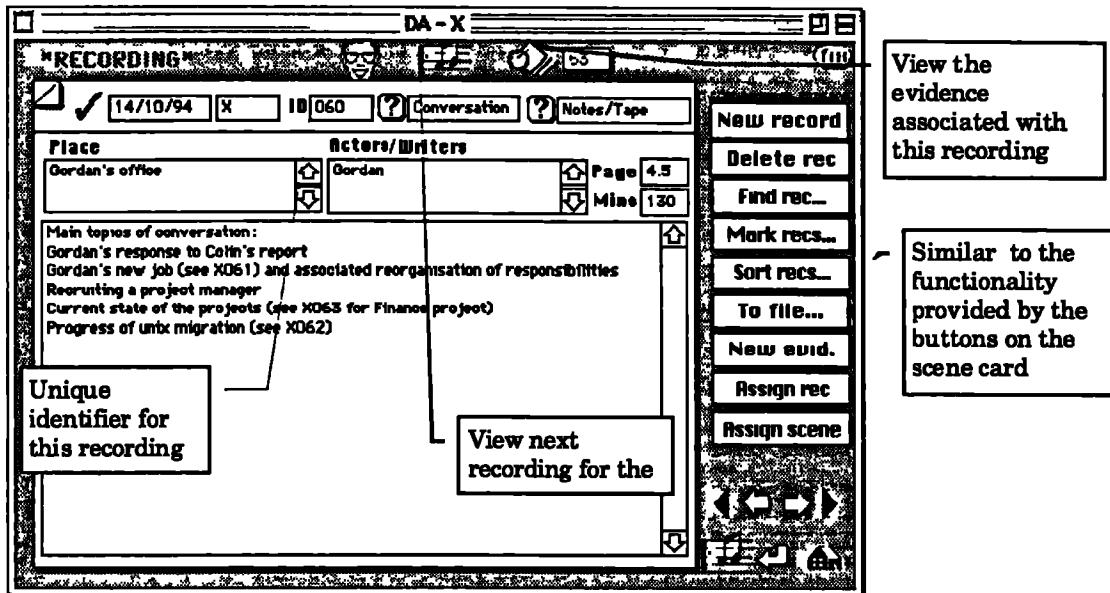


Figure C.3 Recording from scene illustrated in figure C.2

There are nine other recordings associated with this scene (four conversations and five documents). The recording card contains a summary of the content, details of those involved and the lengths of any associated notes or tape recordings. This particular recording is for a conversation with Gordan in his office that resulted in a tapes for 130 minutes and 4.5 pages of written notes.

C.1.3 Evidence

The text associated with each recording (transcript or text of case documents) is stored in evidence cards. There are 63 evidence cards associated with the recording shown in figure C.3. The first of these evidence cards is shown in figure C.4.

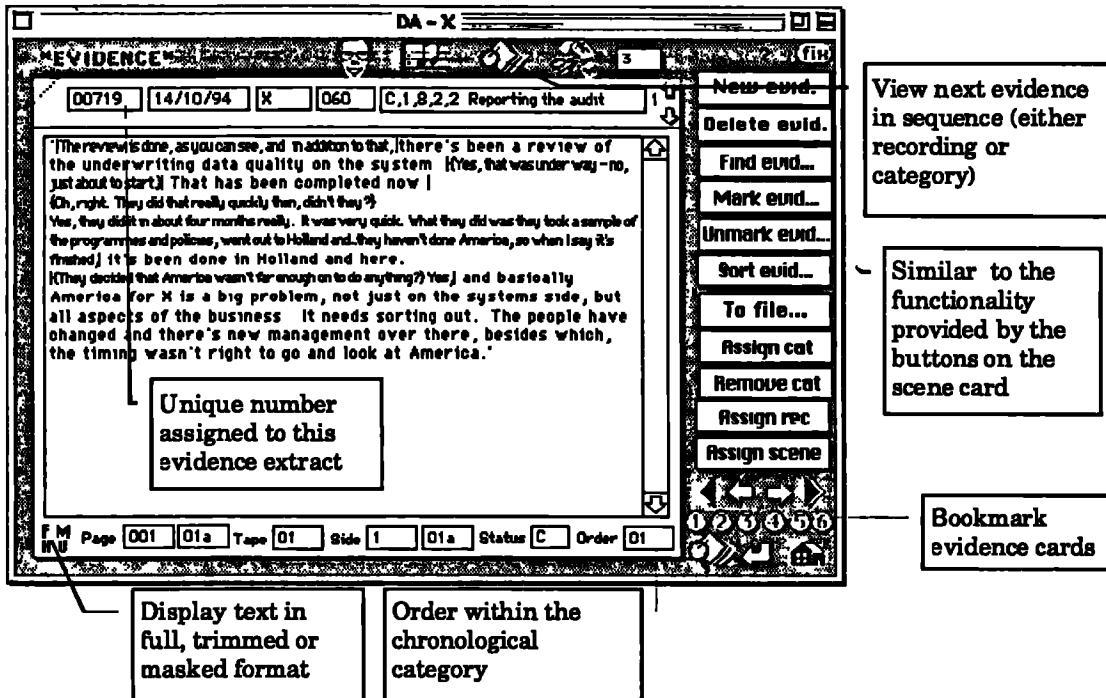


Figure C.4 First evidence item associated with recording X60

The evidence cards were used to enter the text. The position of the text in both notes (or document) and in the tape recording is recorded at the bottom of the card. The evidence cards were also used to create the chronological story. Evidence was categorised chronologically (see C.1.4), ordered and trimmed. The order is included on each card (see figure C.4) with "00" indicating that the card is to be omitted. Trimming was achieved by entering "|" at the start and end of each portion to be trimmed. In figure C.4 the trimmed text is shown in masked format and in figure C.5 in hidden format.

The evidence cards can be categorised on several dimensions. The category for the current dimension of interest is always shown but as figure C.5 shows, all categories can be displayed.

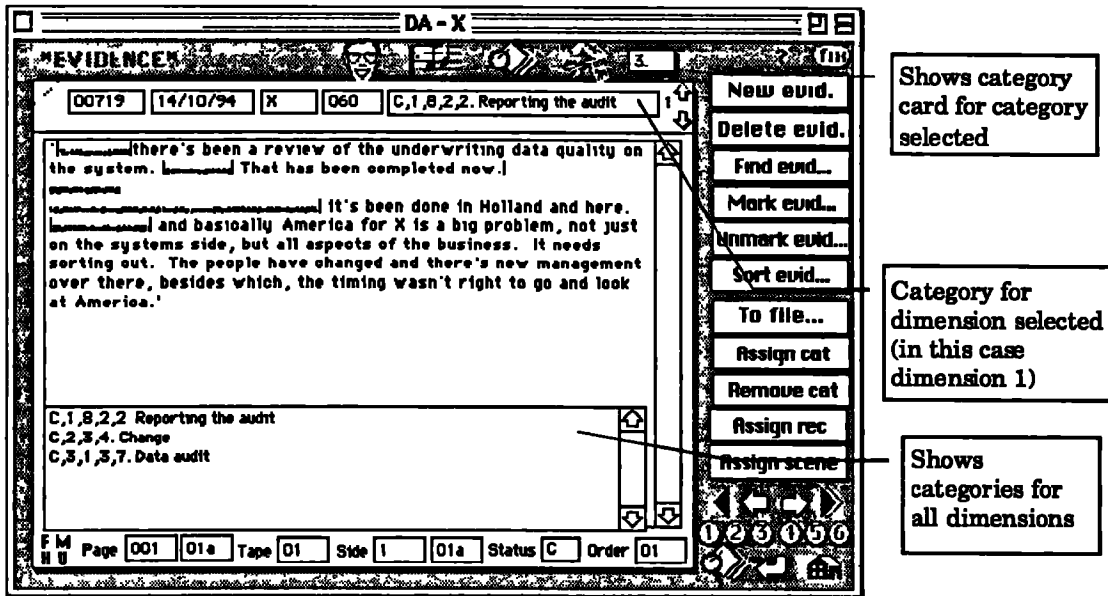


Figure C.5 Same evidence card as figure C.4 with trimmed text hidden and categorisation shown.

C.1.4 Categories

The structure of the category cards is hierarchical except that on the first category card (see figure C.6) the dimensions are defined. Each category is identified by a number indicating its dimension and position in the hierarchy (eg the evidence card in figure C.5 is categorised to the story category C,1,8,2,2.). In addition to the normal buttons provided on each of the cards, the category cards can be moved to anywhere in the hierarchy. When a category is moved all sub-categories and evidence are automatically realigned.

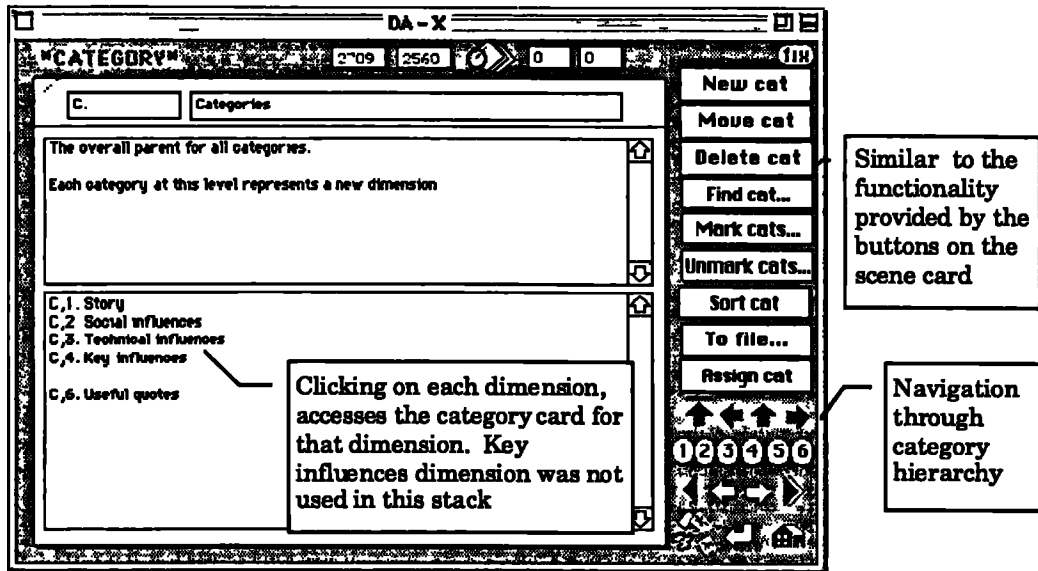


Figure C.6 Main category card for analysis stack

Each category has a memo associated with it describing the category. The memos for the story categories became chapter 6 and the category and sub-categories became the headings and sub-headings in the chapter as illustrated in figure C.7.

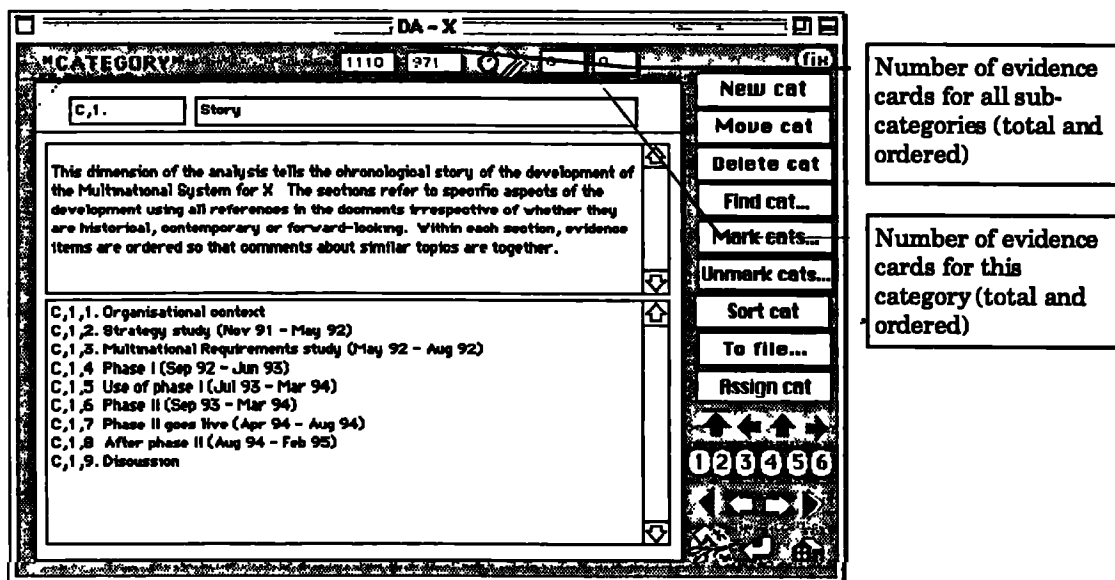


Figure C.7 Top level category card for dimension 1 – chronological story

The analysis stack contains a number of other specialist cards that summarise the content, provide rich text format preferences and allow the web pages to be generated.

C.2 SYNTHESIS STACK

The synthesis stack was used for:

- entering the memos synthesizing the impact of the direct social and technical influences on the software development
- constructing the commentary by categorising and ordering the memo cards and writing the commentary in the category memos
- categorising the commentary according to the key influences using category cards and describing the influences using memos
- developing categories and memos to describe the development in terms of the organisation, people and timing of events
- exporting to Word the commentary for chapter 7 and the discussion of key influences in chapter 8
- constructing and exporting organisation, people, time line, events, commentary, memos and help to the web-site

The synthesis stack was constructed initially as a copy of the analysis stack and so shares similar functionality. The following sections just describe the additional functionality added to the synthesis stack.

C.2.1 Memos

The memo cards have the same functionality as evidence cards in the analysis stack except that where a memo references an evidence extract a hyperlink is constructed to the relevant card.

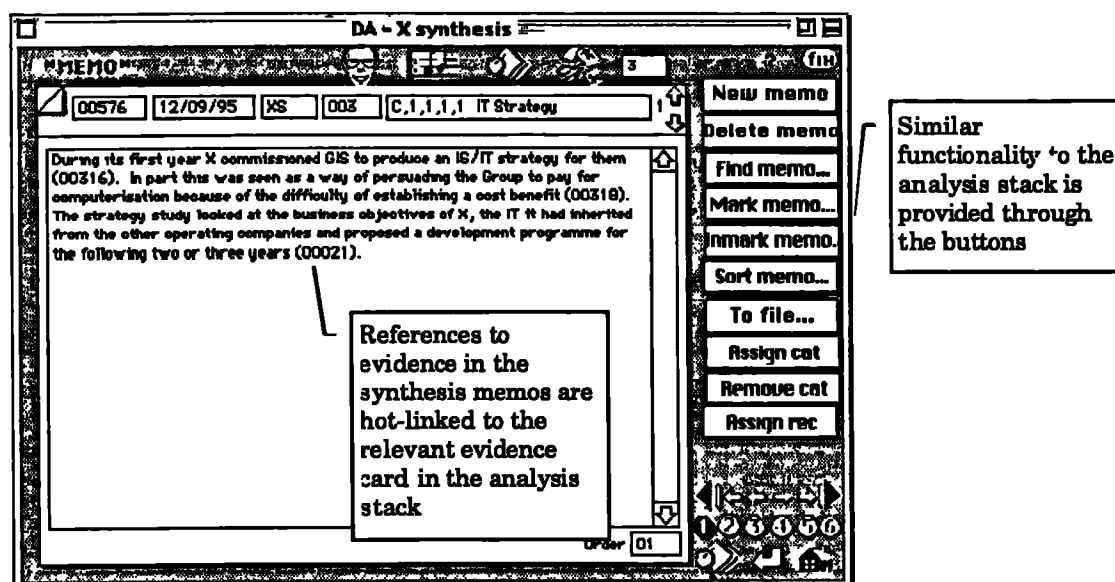


Figure C.8 Memo card forming part of the commentary in the synthesis stack

Memo cards were written as part of the synthesis but were also used to describe other aspects of the development including a discussion of the key influences. The different uses of the memos are indicated by their categorisation.

C.2.2 Categories

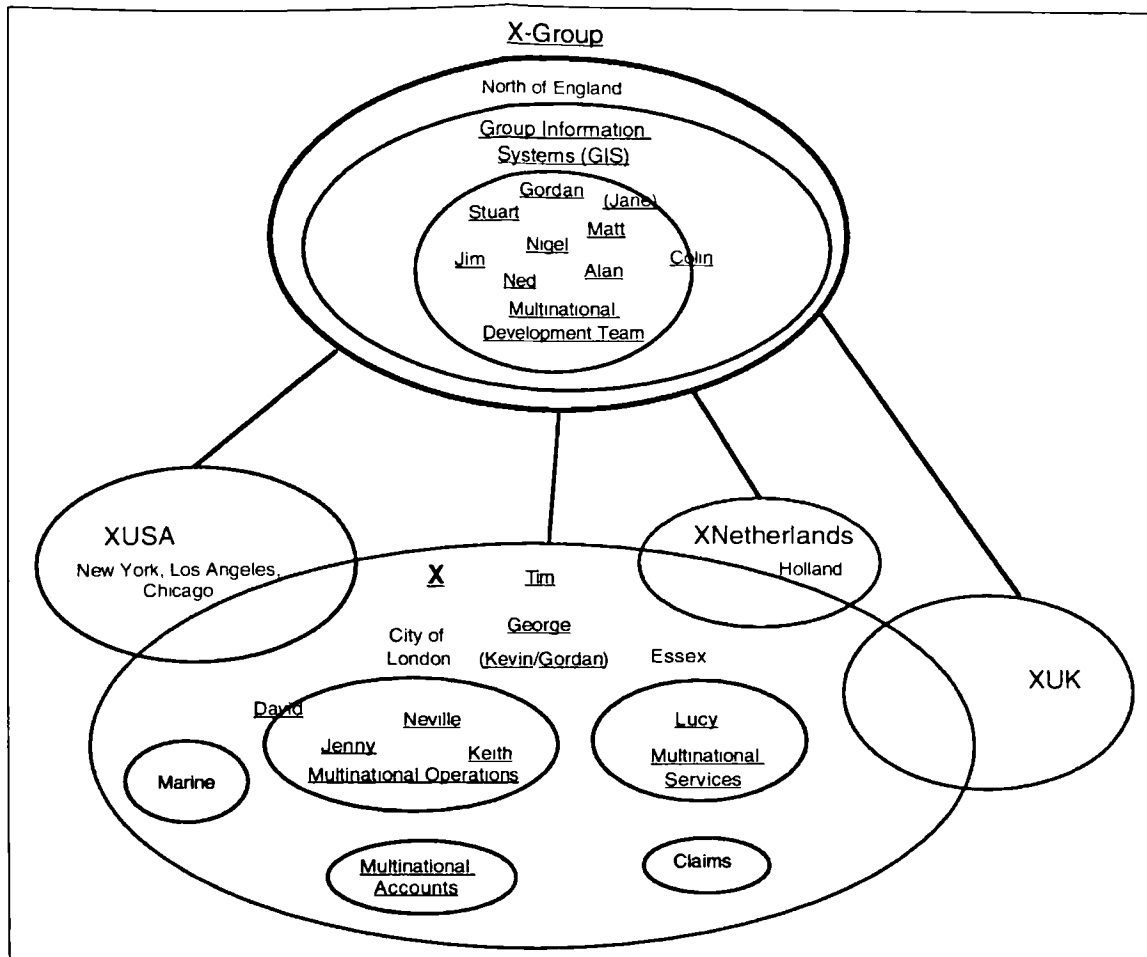
The synthesis memos are categorised on three dimensions: the commentary event and the social and/or technical influence that gave rise to the memo. Categories are also used for key influences and to store information about the case as illustrated in figure C.9.

C.3 PSEUDONYMS STACK

The pseudonyms stack provides a record of the pseudonym used for each person, place and organisation in the cases. A card is not illustrated to protect anonymity.

D Foldouts for chapter 6

Figure D.1 Organisational context of Multinational systems development



- Colin Wrote the IT strategy in 1991/2, undertook the Post Implementation Review of phase II
- David Multinational Business manager until 1993 – seconded to project
- George Finance Director in X from 1993
- Gordan Project leader of Multinational development team from 1992
- Jane Recruited as project manager of Multinational development in 1994
- Jenny Underwriter manager from 1993 – seconded to project part-time
- Keith Underwriter who wrote Q&A system – seconded full-time to project
- Kevin IT manager in X Dec 1993 – Jun 1994
- Lucy Section head of Services from 1992
- Stuart Lead business analyst on Multinational
- Tim Appointed MD when X was created
- Matt, Nigel, Ned, Jim, Alan - programmers

Figure D.2 Time line of Multinational systems development

Year		Organisational events	Development events	Visits
1991		X created	Q&A system	
1992	Q1	MN Services created GIS created		
	Q2	David seconded to project Gordan leads project	IT Strategy Study	
	Q3	Lucy appointed to Services	Requirements Study	
	Q4			
1993	Q1			
	Q2	Jenny replaced David	Phase I installed	CU
	Q3	George appointed		NEng
	Q4	Technical team leader left Kevin appointed IT manager	Phase I review Marine project started	City
1994	Q1	Manny joins Marine team	Phase II installed	NEng
	Q2	Kevin resigned	Fin. Acc. sub-system started Unix migration agreed German standalone installed City requirements discussed	City Essex NEng
	Q3	Database architect left	Phase II review McDonalds system developed Client Services investigated	CU City
	Q4	Gordan & Karl replace Kevin Neville given business resp. Project manager recruited Colin appointed to Fin. Acc.	Data audit Unix installed in UK Process Automation invest.	NEng CU
1995	Q1		Unix installed in US Minor enhancements planned New standards developed New architecture planned US relaunch	City NEng

Key
CU Open University visit
NEng North of England visit
City City of London visit
Essex Essex visit

E Key influences

Figure E.1 shows the critical (⊕), important (✓) and potentially important (?) influences identified in X and critical and important influences identified in the other cases as discussed in section 8.3. The phases of development are indicated in brackets (1-Conception, 2-Construction, 3-Delivery). Where an influence had different impacts at different phases the highest impact is shown.

Figure E.1 Effects of key influences in cases

Key influences	X	V	W	Y maint & blue sky	Z
Individual					
Vision & drive	<ul style="list-style-type: none"> David is driving force (1,3) Who holds the vision (1) Lack of a champion (3) 	<ul style="list-style-type: none"> Chair of IT committee (1) 	<ul style="list-style-type: none"> Robert wants a laptop (1) 	<ul style="list-style-type: none"> Keith (1,2) 	<ul style="list-style-type: none"> Paul has vision for whole project but is it realisable? (1,2) Is vision shared by sponsor? (1,2)
Ability	<ul style="list-style-type: none"> Developer's technical ability valued (2) User's aptitude (3) 	<ul style="list-style-type: none"> Developer ability (2) 		<ul style="list-style-type: none"> Oscar respected despite problems (2) 	
Experiences	<ul style="list-style-type: none"> Previous experience used: David, Keith, Colin & Jane (1,2) Lack of experience unimportant: Alan (2) Lack of experience a problem: Kevin, Stuart (2) 	<ul style="list-style-type: none"> Delivered despite inexperience (2) Not enough Oracle experience (2) Systems experience of users matters (2) 	<ul style="list-style-type: none"> Couldn't visualise applications (1) 	<ul style="list-style-type: none"> Lack of Windows expertise (1) 	<ul style="list-style-type: none"> Users not able to fully contribute when there is nothing tangible, no sense of what might be practical (2) Attitude to marketing – not sufficiently questioning (2) Paul does not understand technicalities (2) Sponsor is new to organisation (2)

Impact of influence: - critical, - important, - potentially important. Phases: 1 - conception, 2 - construction, 3 - delivery

Key influences	X	V	W	Y maint & blue sky	Z
Personal qualities	<p>Important but not discussed as much</p> <p>★ David strong-willed (1,2)</p> <p>✓ Jane tough (2)</p> <p>★ Stuart lack of authority (2)</p> <p>? Seniority (2)</p>	<p>★ Yolande strong willed (2)</p> <p>★ Holly doesn't like managing people (2,3)</p>		<p>★ Neville (2,3)</p> <p>✓ Keith's family commitments (2)</p> <p>✓ Oscar (2)</p>	<p>✓ Implementation manager is a people type job (2,3)</p> <p>★ Paul sticks his head out (1,2)</p> <p>✓ No spiritual bonding with sponsor (1,2)</p>
Relationships	<p>★ David & Gordan good (2)</p> <p>✓ George & Terry poor (2)</p>				
Organisation					
Structure & systems	<p>★ Establishment of X (1,2)</p> <p>★ Physical dispersion (1)</p> <p>★ Set up of Multinational Services (1)</p> <p>★ Existing systems (1,3)</p> <p>★ Who is responsible for data loading? (3)</p>	<p>★ Whose system – two departments (2)</p>	<p>★ Size of company (1)</p>	<p>✓ Family business (1)</p>	<p>★ Decision-making (1,2)</p> <p>✓ Relationship between faculties & centre (2)</p>

Key influences	X	V	W	Y maint & blue sky	Z
Culture	<ul style="list-style-type: none"> ✱ Senior management (1,2,3) ✱ Image of computers (1) ✱ Underwriters & Services (3) 	<ul style="list-style-type: none"> ✓ People regarded as important leading to ill-defined responsibilities (2) 		<ul style="list-style-type: none"> ✱ At heart still a small R&D company of fanatics designing over a curry (1,2) 	<ul style="list-style-type: none"> ✓ Culture in IT department (1,2) ✱ Admin/academics/clerks differences (2) ✓ History not shared with Business School (1)
Organisational learning	<ul style="list-style-type: none"> ✓ Impact of previous phases (1,2) 				
External relationships	<ul style="list-style-type: none"> ✱ Relationship with Sybase (2) 			<ul style="list-style-type: none"> ✱ Dependency on Acorn (1,2,3) 	<ul style="list-style-type: none"> ✓ Relationship with method consultancy (1)
Method					
IT Strategy	<ul style="list-style-type: none"> ✓ Formality to get money (1) ✓ Scope ignored (1) ✱ Set timescale & budget (1) 		<ul style="list-style-type: none"> ✓ Result of PQM partially ignored (1) 		<ul style="list-style-type: none"> ✓ Some inflexibility through sticking to strategy (2) ✱ Ritual significance (1) ✓ Set areas but detail not used (1,2)

Impact of influence: ✱ - critical, ✓ - important, ? - potentially important. Phases: 1 - conception, 2 - construction, 3 - delivery

Key influences	X	V	W	Y maint & blue sky	Z
Methodology	<ul style="list-style-type: none"> ✓ Thorough choice of tools (1) ✱ Phasing & method (1) ✱ Achieved a lot (2) ✱ Catalyst for change (2) ✱ Commitment (2,3) ✓ Career enhancement for developers & users (2) ✱ Difficulty stopping (2) ✱ Requirements problems (2) ✓ Poor internal design (2) ✓ Poor documentation & training (2,3) ✓ Mixed quality of testing (2) ? Existing approaches (1) 	<ul style="list-style-type: none"> ✓ Evolutionary prototyping didn't deliver (2) 		<ul style="list-style-type: none"> ✓ Informal (1,2) 	<ul style="list-style-type: none"> ✱ Mixed reactions to workshops (2) ✓ With a clean sheet of paper only think of the obvious (2) ✱ No allowance for business change in method (1,2) ✓ Lack of flexibility (2) ✓ Tools provide a common language (2) ✓ Too much detail in analysis (2) ✓ Career implications for users & developers (2) ✱ Adapted methodology (1,2,3)
Installation	<ul style="list-style-type: none"> ✱ Poor data (3) ✓ Limited usefulness (3) ✓ Limited development of reports (2) 				<ul style="list-style-type: none"> ✓ Research degree data problems (3)

Impact of influence: ✱ - critical, ✓ - important, ? - potentially important. Phases: 1 - conception, 2 - construction, 3 - delivery

Key influences	X	V	W	Y maint & blue sky	Z
Project management	<ul style="list-style-type: none"> ★ Pressure to deliver (2) ★ No cut-off point (2) ★ Loss of momentum (2) ★ Second system syndrome (2) 	<ul style="list-style-type: none"> ★ Difficult to balance new work & improvements (2) 			<ul style="list-style-type: none"> ✓ Illusion of control (2) ★ PRINCE needed tailoring (2)
Quality assurance	<ul style="list-style-type: none"> ✓ Ineffective at detecting problems (2) ✓ Delays in implementing recommendations (2) ? Sidelined (2) 				<ul style="list-style-type: none"> ✓ PRINCE does not necessarily provide QA (2)
IT profession					
Personality	<ul style="list-style-type: none"> ★ Definite preferences for specific technologies eg PC vs Unix (1) ✓ Expectations (2) ✓ Interests (2) ✓ Definition of success (2) ★ Under-estimate problem (2) 			<ul style="list-style-type: none"> ✓ Ron can't afford to let go code (2) ★ Technical talk – mystique (1,2,3) ✓ Social characteristics (2,3) 	<ul style="list-style-type: none"> ✓ Impossible to constrain the person who leads the design team (2) ★ IT staff are considered 'Scum of the earth' (1,2) ✓ User representatives go native (2)
Management	<ul style="list-style-type: none"> ✓ Difficult to manage (2) 			<ul style="list-style-type: none"> ✓ Need a burn out period (2) ✓ Always got things to do (1,2) 	<ul style="list-style-type: none"> ✓ "Techies only respect techies" (2)

Impact of influence: ★ - critical, ✓ - important, ? - potentially important. Phases: 1 - conception, 2 - construction, 3 - delivery

Key influences	X	V	W	Y maint & blue sky	Z
Recruitment	? Difficult (2)			✓ Can't find a hybrid to manage engineers (2,3)	
Belief					
Developers	⚡ Achieve a lot (1,2)	✓ Nathan committed (2)		⚡ Oscar's progress on print server (2)	✓ Reduce tensions (2)
Users	✓ Reactions to bugs (2) ⚡ Loss of confidence caused by bugs (2,3)	⚡ Short Courses not taking responsibility for data (3)			
Resources					
Staff	✓ Users too busy (2,3) ⚡ Too long to recruit (2)	⚡ Delays & difficulties in recruitment of IT manager & IT staff (2)		✓ Engineer recruitment difficulties (2)	✓ Difficulties in seconding users (1,2)
Deadlines	? Reduced functionality (2) ✓ Poor design (2)				
Money	✓ Came in under budget from not using Unix (1)		⚡ Don't expect to spend money on IT (1)		✓ Cost benefit regarded with scepticism (2)
Equipment		⚡ Machine late but was it ordered late? (2)			

Impact of influence: ⚡ - critical, ✓ - important, ? - potentially important. Phases: 1 - conception, 2 - construction, 3 - delivery

Key influences	X	V	W	Y maint & blue sky	Z
Domain					
Problem	⚡ Complexity (1,2,3) ⚡ Global (1) ✓ Interfacing (1) ✓ Unclear functionality (2)	✓ Billing complexity (2)		✓ Driven by school year (1,2,3) ✓ Can only test on site (2)	⚡ Size of project (1,2)
External factors	✓ State of market (1)			⚡ School funding & influence of growth in PCs (1)	⚡ Changes in HE (1)
Technology					.
Capabilities	✓ Increase in power of PCs (1) ✓ New technologies (1,2) ✓ Power & limitations (2) ✓ Impact on design (2)	✓ Let down by reporting tool (2,3) ⚡ Instability of Oracle (2) ⚡ Integration problems (2) ⚡ Performance (3) ⚡ Aging equipment (1)		⚡ Changing fast (1)	✓ Aging equipment (1)
Working system	⚡ Highlights limitations (1)				

Impact of influence: ⚡ - critical, ✓ - important, ? - potentially important. Phases: 1 - conception, 2 - construction, 3 - delivery

Key influences	X	V	W	Y maint & blue sky	Z
Design	☆ System limited use to underwriters or Services (3) ✓ Validation (3)			✓ Little visible evidence of internal design (2)	✓ Defined at macro level but difficulties converting to minutiae (2)
Change					
External	✓ Designed for change (1) ? Schedules (2)				☆ Integral part of system (1,2)
System induced	✓ User resistance (3)	✓ Not treated as change (3)			☆ Agent for change (3)
Vision	✓ Loss of momentum (2)	☆ No drive at the top (2,3)			✓ Did lack of vision on some areas delay their start? (1)
Communication	? Schedules (2) ? Understanding system (3)	✓ Loss of commitment (3)			☆ Problems despite newsletter, road-shows, workshops (2) ✓ Difficult to get on board ** (2)
Chance			☆ Donald breaks his leg (1)	☆ Sidetracks (1,2)	✓ Chance remark led to changes in analysis (2)

Impact of influence: ☆ - critical, ✓ - important, ? - potentially important. Phases: 1 - concentration